

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
ESCUELA DE POSGRADO



Una investigación sobre arquitecturas que soportan la composición de  
servicios web semánticos : Una Revisión Sistemática

Tesis para obtener el grado de Magíster en Informática con Mención en Ingeniería de Software que  
presenta

Albert Gustavo Quispe Yalle

Dirigida por Dr. Héctor Andrés Melgar Sasieta

San Miguel, 2018

# Agradecimientos

A mi madre, por su apoyo constante e imperecedero, que permitió superar las dificultades que aparentaban no tener solución.

A mi asesor, Dr. Andrés Melgar, por su valioso tiempo y su vasta experiencia en este tipo de investigación, que se convirtió en un aporte significativo que permitió finalizar en buen término la investigación.

A mis amigos y a las personas cercanas, por su interés mostrado en el progreso de la presente investigación que se convirtió en una fuente de motivación.



# Resumen

En la actualidad muchas instituciones están migrando sus procesos de negocio de manera que se encuentren soportados por servicios web para facilitar la interoperabilidad con otras instrucciones y una mejor reutilización de sus componentes, sin embargo la manera sintáctica como se describen los servicios web en la actualidad presentan sus limitaciones cuando se desean hacer actividades automatizadas de descubrimiento, composición e invocación de los servicios web más convenientes. Estas limitaciones están siendo cubiertas con la combinación de conceptos de la web semántica y de los servicios web existentes dando lugar a los servicios web semánticos (SWS). Pero aún existen desafíos que deben resolverse tal como la composición de los servicios web semánticos, este problema se vuelven más críticos sobre todo en las instituciones financieras cuyos procesos de negocio están compuestos por más de un servicio.

El objetivo de este trabajo es llenar ese vacío mediante una revisión sistemática de la literatura existente acerca de las arquitecturas planteadas en la composición de servicios web semánticos. Este documento tiene tres propósitos: en primer lugar, proponer una definición y discusión de las arquitecturas de servicios web compuestos, obtener una clasificación de las arquitecturas en base a sus componentes. En base a clasificación, se identifican y discuten necesidades de investigación futuras.

## Palabras clave

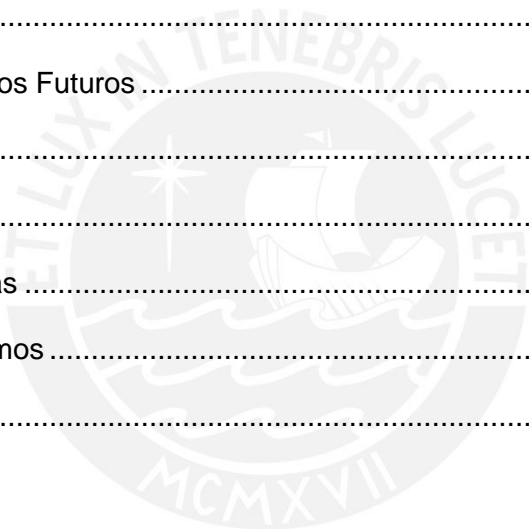
Servicios web, semánticos, arquitectura, seguridad, composición.

# Contenido

Capítulo 1. ....	1
Introducción.....	1
Capítulo 2. ....	3
Marco Conceptual .....	3
2.1    Metadato .....	3
2.2    Ontologías .....	4
2.3    Servicios.....	5
2.4    Composición de servicios web.....	6
2.5    La web semántica.....	7
2.6    El lenguaje XML .....	8
2.7    Lógica de Descripción .....	9
2.8    Servicios Web Semánticos .....	9
2.9    Enfoques de servicios web semánticos .....	9
2.10    Composición de servicios web semánticos.....	12
2.11    Seguridad de los servicios web semánticos.....	13
Capítulo 3. ....	14
Metodología .....	14
3.1    Revisión sistemática de la literatura.....	14
3.2    Identificación de una necesidad para una revisión.....	15
Capítulo 4. ....	16
Protocolo de la revisión .....	16
4.1    Preguntas de búsqueda.....	16

4.2	Estrategia de la búsqueda .....	17
4.2.1	Derivando los términos de búsqueda.....	17
4.2.2	Definiendo las cadenas de búsqueda .....	17
4.2.3	Recursos de búsqueda.....	19
4.2.4	Definiendo cadenas de búsqueda por cada motor de búsqueda.....	19
4.2.5	Proceso de búsqueda.....	21
4.2.6	Documentando el Proceso de búsqueda .....	22
4.3	Criterio de la evaluación de la calidad del estudio .....	22
4.3.1	Criterio de inclusión y exclusión para la selección del estudio .....	23
4.3.2	Lista de aseguramiento de calidad .....	24
4.3.3	Estrategia de extracción de Datos .....	25
4.3.4	Formulario de Extracción de Datos.....	25
4.3.5	Procedimiento de Extracción de Datos .....	27
Capítulo 5.	.....	28
Ejecución	.....	28
5.1	Búsqueda en las Base de datos .....	28
5.2	Búsqueda primaria.....	30
5.3	Búsqueda secundaria .....	33
Capítulo 6.	.....	38
Síntesis y Análisis de Datos	.....	38
6.1	Hechos generales de los estudios seleccionados.....	38
6.1.1	Resultados de la listas de verificación de aseguramiento de calidad.....	38
6.1.2	Instituciones y Países de la investigación .....	39
6.1.3	Campos de aplicación y problemas que resuelven .....	42
6.1.4	Tipo de Arquitectura .....	43
6.2	Respondiendo las preguntas de investigación.....	45

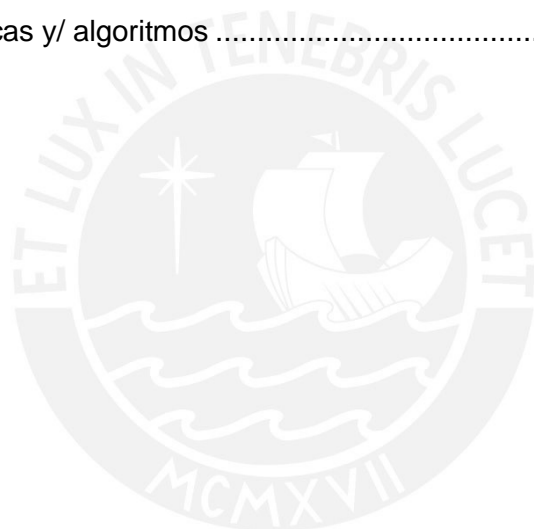
6.2.1	¿Cuáles son los módulos de la arquitectura de los servicios web semánticos compuestos? .....	45
6.2.2	¿Cómo se utilizan las ontologías en una arquitectura de servicios web semánticos compuestos? .....	57
6.2.3	¿Cómo soportar las capacidades de seguridad en una arquitectura de servicios web semánticos compuestos? .....	65
6.2.4	¿Cuáles son las técnicas de composición en una arquitectura de servicios web semánticos compuestos? .....	70
6.2.5	¿Cómo utilizar la inteligencia artificial en una arquitectura de servicios web semánticos compuestos? .....	78
Capítulo 7.	.....	83
Conclusiones y Trabajos Futuros	.....	83
Apéndice	.....	85
I. Lista de Módulos	.....	85
II. Lista de Ontologías	.....	95
III. Lista de Algoritmos	.....	98
Bibliografía	.....	102



# Lista de Tablas

Tabla 4.1 Palabras clave identificadas de las preguntas de búsqueda. ....	17
Tabla 4.2 Sinónimos y términos relacionados a las palabras claves identificados en la tabla 4.1 ....	18
Tabla 4.3 Cadena de búsqueda adaptada a la base de datos SCOPUS .....	20
Tabla 4.4 Cadena de búsqueda adaptada a la base de datos IEEE .....	20
Tabla 4.5 Cadena de búsqueda adaptada a la base de datos ACM DL .....	21
Tabla 4.6 Formulario de Extracción de data.....	27
Tabla 5.1 Registro de la ejecución de las búsquedas .....	29
Tabla 5.2 Número de búsqueda por cada cadena de búsqueda y recurso de búsqueda .....	29
Tabla 5.3 Resumen de la primera búsqueda .....	32
Tabla 5.4 Resumen de la búsqueda secundaria .....	33
Tabla 5.5 Estudios seleccionados al final de la segunda búsqueda.....	36
Tabla 6.1 Resultado de la lista de verificación de aseguramiento de calidad .....	39
Tabla 6.2 Cuadro resumen de estudios por país.....	40
Tabla 6.3 Cuadro detallado de estudios por país .....	41
Tabla 6.4 Campo de aplicación.....	42
Tabla 6.5 Estudios por tipo de arquitectura.....	43
Tabla 6.6 Lista de módulos .....	49
Tabla 6.7 Patrón de Arquitectura. ....	51
Tabla 6.8 Estilos de arquitectura.....	51
Tabla 6.9 Nivel de automatización .....	52
Tabla 6.10 Nivel de dinamismo.....	53
Tabla 6.11 Lenguaje de Ontología.....	58
Tabla 6.12 Etapa en la que interviene la ontología .....	60

Tabla 6.13 Propósito de la ontología.....	62
Tabla 6.14 Etapa en la que interviene los requerimientos de seguridad .....	66
Tabla 6.15 Componente de Seguridad .....	68
Tabla 6.16 Campo de aplicación.....	71
Tabla 6.17 Lista de técnicas .....	75
Tabla 6.18 Rol de la Inteligencia Artificial.....	79
Tabla 6.19 Etapa en la que interviene.....	81
Tabla 0.1 Lista de módulos de la arquitectura.....	94
Tabla 0.2 Lista de ontologías .....	97
Tabla 0.3 Lista de técnicas y/ algoritmos .....	101





# Lista de Figuras

Figura 5.1 Proceso de búsqueda .....	30
Figura 5.2 Tablas Dinámicas. ....	31
Figura 5.3 Herramienta StArt Criterios de Inclusión y de Exclusión.....	31
Figura 5.4 Gráfico estadístico .....	32
Figura 5.5 Gráfico estadístico .....	34
Figura 5.6 Artículos clasificados por año.....	37
Figura 6.1 Estudios por país .....	40
Figura 6.2 Campo de aplicación .....	43
Figura 6.3 Tipo de Arquitectura.....	44
Figura 6.4 Componentes de arquitectura.....	50
Figura 6.5 Patrón de arquitectura .....	51
Figura 6.6 Nivel de automatización .....	53
Figura 6.7 Nivel de Dinamismo .....	54
Figura 6.8 Arquitectura de servicios web compuestos semánticos.....	55
Figura 6.9 Lenguaje de ontología.....	58
Figura 6.10 Etapa en la que interviene la ontología .....	60
Figura 6.11 Propósito de la ontología .....	63
Figura 6.12 Etapa en la que interviene .....	66
Figura 6.13 Componente de seguridad.....	69
Figura 6.14 Campo de aplicación .....	72
Figura 6.15 Técnicas utilizadas en la arquitectura .....	76

Figura 6.16 Rol de la inteligencia artificial .....80

Figura 6.17 Etapa en la interviene .....81



# Capítulo 1.

## Introducción

Antes de que apareciera el concepto de servicio web, las aplicaciones se basaban en componentes de software propietarios, con un alto acoplamiento con la plataforma sobre los cuales se había desarrollado, esto suponía un gran problema cuando se requería una interacción entre los componentes basados en distintas soluciones como por ejemplo soluciones basadas en diferentes lenguajes de programación. En respuesta a las nuevas necesidades propias de un entorno abierto y de intercambio de información entre varias aplicaciones que pertenecen a diferentes instituciones, se desarrolló un conjunto de estándares para que fueran aceptadas por la comunidad web. Los organismos tales como OASIS(Organization for the Advancement of Structured Information Standards) y la W3C (World Wide Web Consortium) establecieron los siguientes estándares sobre los cuales se basaría la tecnología de servicio web :WSDL (Web service Description Language),SOAP (Simple Object Access protocol) y UDDI ( Universal Description ,Discovery and Integration ) . Sin embargo esta manera sintáctica de describir el servicio web produjo una serie de limitaciones cuando se deseaba automatizar las tareas de descubrimiento, composición e invocación de servicios web debido a que las descripciones de los servicios no eran legibles por un computador. En el 2001 Tim Berners-Lee en su famoso artículo sobre la web semántica [1] proporcionaba los fundamentos sobre los cuales los agentes tenían la posibilidad de entender, analizar y razonar las páginas de la web de modo tal que puedan satisfacer las necesidades de los usuarios, siendo los componentes principales sobre los cuales se basa la web semántica: Ontología, RDF(Resource Description Framework) y XML(Extensible Markup Language). Ante la aparición de tales conceptos se propuso resolver la limitación señalada anteriormente, dando esto origen a los servicios web semánticos.

En un principio las soluciones propuestas abordaban los problemas en los requerimientos funcionales sobre servicios atómicos, pero para que una solución sea aplicable en el mundo real, las arquitecturas deben soportar tareas complejas en donde se requiere más de un servicio atómico para responder una solicitud de negocio. Por tal motivo urge realizar investigaciones sobre la composición de servicios web semánticos para poder lograr su aplicación masiva en el mundo real.

El objetivo de este trabajo es llevar a cabo una revisión sistemática de las arquitecturas existentes que soportan la composición de servicios web semánticos de modo tal que tengan una aplicación

práctica. Luego teniendo como base tales estudios se podrá hacer evaluación y síntesis de la información tomando en cuenta las preguntas de investigación.

El documento está organizado de la siguiente manera: en el capítulo II se presenta el marco conceptual donde se presenta una breve descripción de los principales conceptos relacionados a los servicios web semánticos. En la sección III se define la metodología a utilizar en la investigación. En la sección IV se define el Protocolo de Revisión para establecer las preguntas de investigación, los criterios de inclusión y exclusión y la lista de aseguramiento de calidad. En la sección V se ejecuta el proceso de búsqueda, seleccionando los estudios por medio del protocolo de revisión. En la sección VI se hace el análisis y síntesis de datos en donde se procede a responder las preguntas de investigación utilizando el formulario de extracción de datos y por último en la sección VII se presenta las conclusiones y trabajos futuros.



# Capítulo 2.

## Marco Conceptual

### 2.1 Metadato

El metadato es una parte fundamental de un dato, debido a que brinda una descripción útil y significativa del dato subyacente que puede ser utilizada para satisfacer los requerimientos de diferentes tipos de usuarios. La información capturada por el metadato permite conocer por ejemplo como un dato es almacenado, administrado y el contenido semántico de este. Debido a que el metadato en general es más pequeño en magnitud que el dato al cual describe, su utilización da como resultado un rendimiento más eficiente en procesos tales como búsqueda e inferencias por citar algunos procesos. Esta eficiencia se basa en que las descripciones del metadato son abstracciones independientes de los detalles de representación y también permiten describir la información de dominio del dato al cual se relacionan. Así mismo es recomendable tener múltiples niveles de metadatos en una manera recursiva.

Según el autor [2] hace referencia a una clasificación del metadato tomando como criterio de clasificación el nivel de abstracción con que el metadato describe el dato subyacente. Los metadatos se pueden clasificar en forma sintáctica, estructural y semántica los cuales se describen a continuación:

- a. Metadato sintáctico. Cuando el metadato se centra en detalles del dato subyacente tales como fecha de creación, lenguaje utilizado, tamaño de datos. Esta información es utilizada para clasificar el dato subyacente.
- b. Metadato estructural. Cuando el metadato se centra en la estructura del documento, es decir si sigue un esquema XML o XSL por citar algunos ejemplos. Esta información es utilizada cuando se intenta almacenar, procesar o mostrar un documento.

- c. Metadato semántico. Cuando el metadato se centra en información relevante del contexto con relación al dominio al que pertenece el documento. Es una información significativa e interpretable por humanos y por el software.

## 2.2 Ontologías

La palabra ontología tiene diferentes significados de acuerdo al campo de aplicación en el cual se utiliza. En el campo de la filosofía la Ontología es una disciplina en el cual se estudia la naturaleza y la estructura de la "realidad", independientemente que el objeto exista o sea algo ficticio. Aristóteles trata sobre este término en su libro Organon [3] .

En el campo de la inteligencia artificial, primero Neches y sus colegas definen una ontología de la siguiente manera "Una ontología define los términos básicos y las relaciones que componen el vocabulario de un área temática, así como las reglas para combinar términos y relaciones para definir extensiones al vocabulario" [4]. Esta definición indica que para construir una ontología es necesario identificar y definir los términos básicos, las relaciones entre ellos y las reglas que permiten combinar los términos y las relaciones para generar nuevos términos dando por resultado un enriquecimiento del vocabulario. Posteriormente en 1993, Gruber dio un concepto de la ontología ampliamente citado: "Una ontología es una especificación explícita de una conceptualización." [4]

En 1995 Guarino y Giaretta redefinieron el significado de ontología dada por Gruber , tomando en cuenta 7 posibles significados en el que se usa el termino de ontología . Dando por resultado dos significados [5] :

- a) La ontología es una teoría lógica representada como un "artefacto diseñado, una base de conocimientos de un tipo especial que se puede leer, vender o compartirse físicamente".
- b) La ontología es un sinónimo de conceptualización se define como "una estructura semántica intencional que codifica las reglas implícitas que restringen la estructura de una parte de la realidad".

Luego esta definición fue refinada por Borst (1997) de la siguiente manera: "Una ontología es una especificación formal de una conceptualización compartida." Posteriormente Studer y sus colegas (1998) se encargaron de fusionar las definiciones de Gruber y Borst: "Una ontología es una especificación formal y explícita de una conceptualización compartida." Se puede definir la

conceptualización como una interpretación estructurada de una determinada área de la “realidad”. El término de “especificación explícita” se refiere a que se definen en forma clara y explícita las entidades y las relaciones entre ellas de un área determinada. El término formal se refiere a que los conceptos identificados deben de ser entendibles por las máquinas. Finalmente el término “compartido” se refiere a que la ontología es un conocimiento aceptado en consenso por un grupo [4] [6].

Los componentes de una ontología son [5]:

- a. **Una clase o concepto.** Es un grupo de entidades a las que se hacen referencia y que comparten características comunes. Por ejemplo pueden ser un tipo de objeto, la descripción de una tarea, función, acción, estrategia, proceso de razonamiento.
- b. **Las relaciones.** Se refiere a la manera en que interactúan los conceptos o clases en un determinado dominio.
- c. **Las funciones.** Son casos particulares de las relaciones.
- d. **Los axiomas.** Son condiciones lógicas que capturan el conocimiento de la ontología y que se asumen como ciertas.
- e. **Las instancias.** Son las ocurrencias en el mundo real de los conceptos o clases. En una instancia los atributos tienen asignado un valor concreto.

## 2.3 Servicios

Se tiene varios significados sobre el término de “servicios” de acuerdo al contexto en donde se utiliza, a continuación se tiene un clasificación del significado de “servicios”

1. **El servicio como actividad.** Esta definición es utilizada por ejemplo en el campo del planeamiento de la inteligencia artificial o de sistemas de agentes. Siendo el servicio una actividad simple o compleja, permitiendo la composición de servicios para soportar la creación de flujos de trabajos inter-organizacionales.
2. **El servicio como un componente de software.** En los sistemas distribuidos, networking y telecomunicaciones un servicio es un componente de software que encapsula una determinada funcionalidad de una aplicación. Por ejemplo un servicio de autenticación encapsula solo la funcionalidad del acceso a un sistema en específico.

3. **El servicio como un intercambio interactivo de intangibles.** En el contexto de la economía, gestión y comercialización, el servicio es considerado como un bien intangible, debido a que se le trata como un acto o rendimiento ofrecida por una persona a otra.

Los principales participantes que interactúan en una prestación de servicio tienen los siguientes roles: El solicitante o cliente del servicio, el proveedor del servicio. En el proceso de descubrimiento se pueden distinguir otros roles tales como: El buscador de servicios que intenta encontrar los servicios que satisfaga la solicitud, el intermediario que interactúa con el proveedor del servicio en representación del cliente, el agente de registro que tiene un registro de los servicios disponibles y los entrega al solicitante.

Hay tres aspectos básicos en los que deben ponerse de acuerdo los participantes: funcional, comportamiento y no funcional. Los aspectos funcionales, describen lo que un servicio puede hacer, los aspectos de comportamiento describen cómo se puede lograr la funcionalidad ofrecida, las propiedades no funcionales capturan restricciones sobre el aspecto funcional y de comportamiento, tales como disponibilidad, calidad entre otras. [7] .

## 2.4 Composición de servicios web

La composición de servicios web se da cuando las necesidades o metas de un cliente requieren la invocación de múltiples servicios para poder satisfacerlos. Esta composición conlleva a problemas de coordinación que deben ser resueltos. Bajo la perspectiva de los servicios existen dos enfoques básicos [7] para la coordinación de servicios Web, los cuales son conocidos como orquestación y coreografía.

a. **Orquestación.** Es un enfoque centralizado, orientada al “conductor”, que se centra en la implementación interna de un servicio. El lenguaje de orquestación más adoptado en la industria es el WS-BPEL (Lenguaje de ejecución de procesos de negocio con servicios web) , en un inicio conocido como BPEL4WS . El WS-BPEL se basa en dos lenguajes de especificación de flujos de trabajo, tales como: El WSFL (Lenguaje de flujos para servicios web) de IBM y el XLANG de Microsoft (Servicios web para el Diseño de Procesos de Negocio). WS-BPEL es un lenguaje complejo que adopta las estructuras de control como por ejemplo: secuencia, mientras, en caso entre otras del XLANG y la propuesta basada en gráficos dirigidos del WSFL. El WSFL , es un lenguaje para describir los flujos de un servicio web , donde cada invocación de un servicio es



representado como una tarea de un flujo de trabajo y los flujos de datos y de control son representados como arcos que relacionan nodos de tareas [8]

*b. Coreografía.* La coreografía se enfoca más en la interacción externa entre los servicios y los clientes que consumen el servicio. La coreografía muestra la composición desde un punto de vista global, donde la interacción de los servicios se describen a través del intercambio de patrones de mensajes para alcanzar posibles metas comunes de negocio. WS-CDL(lenguaje de descripción de coreografías de servicios web) , es una propuesta de coordinación basada en coreografías , especificado por W3C( El consorcio www) . WS-CDL es un lenguaje basado en XML para describir colaboraciones de igual a igual entre los servicios web que participan en una coreografía. El objetivo de especificar coreografías de servicios web es componer la interacción entre cualquier tipo de servicios, con independencia del lenguaje de programación o del entorno que alojan al servicio [9] .

## 2.5 La web semántica

La web semántica considera la web como una amplia fuente de información, que puede ser interpretada por personas y sistemas de aplicación distribuida y heterogénea. La web semántica modela la información con el propósito de compartir y reutilizar el conocimiento que da como resultado búsquedas más exactas de información y de servicios. Así mismo provee la infraestructura y las técnicas necesarias para razonar sobre las descripciones ontológicas de los términos y conceptos, además permite la resolución y mapeo entre ontologías, y la interoperabilidad semántica de los servicios web a través de la identificación de los conceptos semánticamente similares. La web semántica dispone de unos principales componentes construidas sobre tecnologías tales como URI y esquemas XML, los cuales son: RDF, Esquema RDF y OWL (lenguaje de Ontologías Web) [10].

1. **RDF.** La descripción de Recursos (RDF) es un lenguaje que usa la notación XML para descomponer la información en piezas representadas como proposiciones y que se pueden formar en estructuras generales como grafos dirigidos etiquetados. Las proposiciones se representan en la forma de sentencias triples sujeto-predicado-objeto y los términos utilizados son singularmente identificados con las URIs. La ventaja de RDF sobre XML es que soporta la idea de combinar información de múltiples fuentes en la Web y permite hacer inferencias que pueden originar nuevas piezas de información [7].

2. **RDF Schema.** Puede ser usado para especificar vocabularios (clases y propiedades) para usarlos en las descripciones RDF. El esquema RDF hace posible decir que tipo de propiedades pueden ser asociadas a diferentes recursos y cuáles son las relaciones de esas propiedades [7]. El Esquema RDF es un sistema de tipo simple para RDF [11]. Proporciona un mecanismo para definir las propiedades específicas al dominio y las clases de recursos a los que se pueden aplicar esas propiedades. Las primitivas básicas de modelado en un esquema RDF son las definiciones de *clase* y las declaraciones de *subclase* (que en conjunto permiten la definición de jerarquías de clases), las definiciones de *propiedad* y las declaraciones de la *subpropiedad* (para construir jerarquías de propiedad), declaraciones de *dominio* y de *rango* (para restringir las posibles combinaciones de propiedades y clases), y las declaraciones de *tipo* (para declarar un recurso como una instancia de una específica clase). Con estas primitivas se puede construir un esquema para un dominio específico.
3. **OWL.** Es un lenguaje de representación de ontología estandarizado por W3C. En un inicio, dos grandes proyectos trabajaban en forma separada en el desarrollo de dos lenguajes de representación de ontologías. Estos proyectos fueron el lenguaje de Intercambio de ontología (OIL) y el DARPA (DAML). Ambos proyectos se fusionaron en el proyecto de DAML+OIL .OWL se basa conceptualmente en RDFS. En la práctica, se realiza como la extensión del vocabulario de RDF (S). Existen tres variaciones poderosas de OWL: a) OWL Full, OWL DL (Descripción Lógica) y OWL Lite. OWL FULL incluye todas las primitivas del lenguaje y, por lo tanto, representa la variación OWL más potente y completamente compatible con RDF, OWL DL no es completamente compatible con RDF, y está parcialmente restringido pero permite un eficiente soporte de razonamiento, OWL Lite, es la variación menos poderosa y es una sub colección de OWL-DL [12] .

## 2.6 El lenguaje XML

El lenguaje XML permite la representación de documentos estructurados. Permite codificar una amplia variedad de estructuras de datos en una sintaxis clara. XML es un metalenguaje que pertenece a una clase de lenguajes de marcado, siendo los principales componentes de marcado los *elementos*. XML permite que los elementos contengan otros elementos o texto. A pesar de sus fortalezas no permite especificar el uso de los datos ni su semántica, debido a su representación jerárquica y a su propuesta orientado a documentos. Estas debilidades no son

muy útiles para el razonamiento y la representación del conocimiento. No es natural empaquetar la información en documentos cerrados en un entorno abierto, distribuido y evolutivo [7] .

## 2.7 Lógica de Descripción

La lógica de descripción es una familia de formalismos para la representación del conocimiento conceptual. Pueden ser utilizados para dar descripciones estructurales de las entidades. La base de conocimiento de la lógica de descripción está formada por dos componentes: TBox contiene todas las sentencias que describen los conceptos y sus relaciones y RBox que contiene todas las sentencias donde aparecen los individuos. Las descripciones lógicas se utilizan en la web semántica sobre todo para apoyar el diseño de ontologías para mejorar la calidad de las definiciones resultantes [7].

## 2.8 Servicios Web Semánticos

Son servicios web definidos a través de una ontología de servicio que permiten su descubrimiento, ejecución e invocación automática a través de dominios e individuos heterogéneos. Se diferencian de los Servicios Web convencionales debido a que estos sólo proveen descripciones a nivel sintáctico, haciendo difícil a los proveedores y solicitantes la interpretación y representación del significado de las entradas, salidas y de las restricciones de los servicios web. Las infraestructuras de servicios web semánticos se pueden caracterizar en tres dimensiones ortogonales: Actividades de uso, la arquitectura y la ontología de servicio. Actividades de uso definen los requisitos funcionales que se deben soportar. La arquitectura de SWS define los componentes necesarios para llevar a cabo estas actividades. La ontología de servicios describe las capacidades funcionales de un servicio web, las capacidades no funcionales, la información relacionada al proveedor, la información relacionada a las metas o tareas y la definición del conocimiento de dominio [10] .

## 2.9 Enfoques de servicios web semánticos

1. **Enfoque Iris.** El Servicio de Razonamiento de Internet - IRS-II es un framework de servicios web semánticos, que permite a las aplicaciones describir y ejecutar semánticamente servicios web. IRS-II se basa en el framework UPML (Lenguaje de desarrollo de método unificado de resolver problemas), que distingue entre las siguientes categorías de componentes especificados por medio de una ontología adecuada [10] :

- a. Los modelos de dominio. Estos describen el dominio de una aplicación.
- b. Los modelos de tareas. Estos proporcionan una descripción genérica de la tarea que hay que resolver, especificando las entradas, salidas, metas y pre condiciones.
- c. Los métodos para solucionar problemas (PSM). Estos proporcionan descripciones abstractas, independiente de la implementación sobre procesos de razonamiento que pueden ser utilizadas para resolver tareas en un dominio específico.
- d. Puentes. Estos especifican los enlaces entre los diferentes componentes del modelo dentro de una aplicación.

Los principales componentes de la arquitectura IRS-II son el servidor IRS-II, el publicador IRS-II y el Cliente IRS-II, que se comunican a través del protocolo SOAP.

2. **Enfoque SAWSDL.** La especificación SAWSDL permite que anotaciones semánticas pueden ser agregados a documentos WSDL. Los principios clave de diseño para SAWSDL son [13] :

- a. La especificación permite anotaciones semánticas de los servicios sobre el marco de desarrollo existente y extensible de WSDL;
- b. Es agnóstico a lenguajes de representación semántica (ontología);
- c. Permite anotaciones semánticas para los servicios web para el descubrimiento e invocación de servicios web.

La mayor crítica a esta especificación es una simple extensión sintáctica de WSDL a diferencia de OWL-S y WSMO que proponen ontologías de nivel superior.

3. **El enfoque METEORS.** el framework METEORS permite la definición de procesos web comparables a flujos de trabajo. Estos procesos web están compuestos de servicios web. Definiciones abstractas de proceso son usados para definir semánticamente el comportamiento de los procesos .Estas definiciones abstractas consisten en una serie de servicios semánticas,

definiendo particulares operaciones abstractas. Las propiedades funcionales, así como las no funcionales de los servicios son tomadas en cuenta en el proceso de selección. Este marco de desarrollo permite la reconfiguración dinámica en tiempo de ejecución durante todo el ciclo de vida. Así mismo este marco de desarrollo presenta componentes mediadores para tratar servicios heterogéneos. [12]

4. **El enfoque OWL-S.** Se compone de un conjunto de ontologías diseñadas para describir y razonar sobre las descripciones de servicio. OWL-S combina la expresividad de las lógicas de descripción y el pragmatismo que se encuentra en los emergentes estándares de los servicios web, para describir los servicios en forma semántica, y establecido dentro de un formalismo bien definido de tipos de datos. Se compone de tres principales ontologías superiores: El perfil, el modelo de proceso y el grounding. *El perfil* se utiliza para describir los servicios para propósitos de descubrimiento, allí se definen las propiedades funcionales y no funcionales, *Los modelos de procesos* OWL-S describen la composición o la orquestación de uno o más servicios en términos de sus procesos constituyentes. *El Grounding* facilita la ejecución, permitiendo un enlace entre la parte conceptual y la parte física. [10]
5. **El enfoque WSMF.** El framework de modelado de servicios se centra en dos principios complementarios: un fuerte desacoplamiento de los diversos componentes que realizan una aplicación de comercio electrónico; y un fuerte servicio de mediación que permiten a los servicios web comunicarse en una manera escalable. La mediación se aplica a varios niveles: la mediación de estructuras de datos; la mediación de la lógica de negocio; la mediación de los protocolos de intercambio de mensajes; y la mediación de la invocación dinámica del servicio. WSMF consta de cuatro elementos principales: las ontologías que proporcionan la terminología utilizada por otros elementos; los repositorios de metas que definen los problemas que deben ser resueltos por los servicios Web; las descripciones servicios Web que definen los diversos aspectos de un servicio Web; y los mediadores que tratan los problemas de interoperabilidad. WSMF aplicación ha sido asignado a dos proyectos principales: SWWS(servicios web habilita por la web semantica) Y WSMO(Ontología de modelamiento de servicios web) SWWS proporcionará un marco de descripción, descubrimiento y una plataforma mediación para servicios web, de acuerdo a una arquitectura conceptual. WSMO refina el WSMF y desarrolla un lenguaje y ontología formal de servicio para los servicios web semánticos. [10]

## 2.10 Composición de servicios web semánticos

En una composición de servicios la lógica de negocio es implementada por la combinación de funcionalidades de varios servicios. Una composición de servicios tiene que definir el orden en el que este invoca los componentes. En la mayoría de los casos, el orden de invocación puede ser secuencial, de acuerdo a una condición previa requerida por la lógica de negocio y también puede ser en forma aleatoria siempre y cuando el orden es irrelevante para implementar la lógica de negocio. Los enfoques de composición se pueden categorizar como manuales, semiautomáticos y automáticos. En la composición manual, un usuario elige los servicios Web relevantes, las composiciones manuales son difíciles de mantener y consumen mucho tiempo. Con la composición automatizada, el usuario final o el desarrollador de la aplicación especifica una meta y un motor de composición selecciona los servicios adecuados y ofrece una composición transparente al usuario.. Las estrategias de composición se pueden clasificar en no-informadas, heurísticas, evolutivas y enfoques basadas en la calidad de servicio [14] .

- a. **Propuesta no informada.** Es el enfoque más general y directo a la composición de servicios web. Se caracteriza porque no hacen uso de una información previa. Hay muchas maneras de desarrollar algoritmos basados en este enfoque. Debido a que es una estrategia de búsqueda exhaustiva, este método es lento y consume mucha memoria, ya que no utiliza ninguna información adicional sobre el espacio de búsqueda. Esta estrategia es adecuada en repositorios o espacios de búsqueda pequeños.
- b. **Propuesta Heurística.** El uso de información adicional sobre el espacio de búsqueda puede aumentar notablemente la eficiencia. En una búsqueda informada, una buena heurística podría superar considerablemente el tiempo utilizado por el proceso de búsqueda de una propuesta no informada, sin embargo, esta no siempre garantiza conseguir resultados óptimos , debido a que depende de la calidad de conocimiento adicional que se dispone antes de iniciar la búsqueda.
- c. **Propuesta Evolutiva.** Este enfoque define algoritmos de optimización meta-heurística que utilizan mecanismos inspirados en la biología como la selección natural, la mutación, supervivencia del más apto entre otros aspectos. Los dos factores que deben tenerse en cuenta aquí son el tamaño de la composición y el número de parámetros insatisfechos. Se ha encontrado evidencia que el enfoque evolutivo es más lento, pero siempre encuentra la composición correcta para todas las solicitudes.

**d. Propuesta Basada en Calidad.** Este enfoque representa algoritmos que toman en cuenta aspectos no funcionales para lograr la composición. Permite lograr una mejor selección de servicios web que componen un servicio web compuesto.

## 2.11 Seguridad de los servicios web semánticos

La seguridad es un aspecto muy importante a tomar en cuenta en aplicaciones críticas para el negocio. Los lenguajes de ontologías de servicios web tales como OWL-S y WSMO cubren capacidades de seguridad en la descripción de los servicios web semánticos, a continuación se describe la forma como la seguridad es soportada por los dos lenguajes de ontologías:

- a. **OWL-S.** Este lenguaje tiene sub-ontologías: El perfil de servicio, El modelo de servicio y el grounding (informa como acceder al servicio) .Es el perfil del servicio el que contiene parámetros que permiten describir propiedades no funcionales tales: como nombre del servicio, descripción de texto y clasificación de la calidad. [15]
- b. **WSMO.** Este lenguaje propone una ontología de seguridad genérica para definir requerimientos de seguridad y confidencialidad. La desventaja de esta ontología es que además de ser muy genérica no cubre importantes propiedades de seguridad tales como: integridad, no repudio, autenticación, las mismas por omitirse hace difícil implementar la ontología.

# Capítulo 3.

## Metodología

### 3.1 Revisión sistemática de la literatura

Para el presente estudio se utilizó la metodología de la revisión sistemática, debido a que la características expuestas por Kitchenham [16] muestran una investigación más rigurosa , ordenada y reproducible .Entre las características expuestas resalta que toda revisión sistemática se inicia especificando las preguntas de investigación, para luego especificar la estrategia de búsqueda y la documentación respectiva .Donde la definición de los criterios de inclusión y exclusión es importante para evaluar e identificar los estudios más relevantes que puedan responder a las preguntas de investigación en un inicio formuladas.

Una revisión sistemática de la literatura, mencionada por Kitchenham, tiene tres etapas principales: planificar la revisión, realizar la revisión e informar la revisión. Esas fases se presentan con mayor detalle en la siguiente lista, que fue adaptada de la presentada en [16] :

#### **1 Planificación de la revisión**

- a. Identificación de una necesidad de la revisión
- b. Puesta en marcha de una revisión
- c. Especificar las preguntas de Búsqueda
- d. Desarrollar un protocolo de revisión
- e. Evaluar el protocolo de revisión.



## **2 Conduciendo la revisión**

- a. Identificación de la investigación
- b. Selección de los estudios primarios
- c. Estudio de aseguramiento de calidad
- d. Monitoreo y extracción de datos
- e. Síntesis de datos

## **3 Informar la revisión**

- a. Especificar los mecanismos de difusión
- b. Formatear el reporte principal
- c. Evaluar el reporte

### **3.2 Identificación de una necesidad para una revisión**

Los servicios web semánticos aunque es una tecnología muy prometedora para la automatización de las interacciones entre los servicios web, aun presentan problemas que no han sido resueltos y que dificultan su utilización en las transacciones comerciales, según el estudio [17] identifica tres problemas principales que deberían ser resueltos para su utilización, tales como: la composición de servicios, el descubrimiento del servicio y la confianza. Si bien el citado estudio propone algunas soluciones, la presente revisión sistemática tiene como propósito resumir toda la información existente sobre arquitecturas que soportan la composición de servicios web semánticos de una forma metodológica y reproducible, debido a que existe poca profundidad en los estudios relacionados al objetivo de la presente investigación.

# Capítulo 4.

## Protocolo de la revisión

El protocolo de revisión es una parte fundamental de una revisión sistemática. Este está formado por una serie de componentes o actividades para llevar a cabo la revisión. Entre los componentes más importantes se encuentra el establecimiento de las preguntas de investigación, la estrategia de búsqueda, los criterios de aseguramiento de calidad y la estrategia de extracción de datos para registrar la información encontrada [16]. Es necesario que el protocolo de revisión siempre se encuentre en constante evaluación, existiendo la posibilidad de revisarlo de nuevo si se encontrara un problema en su ejecución.

### 4.1 Preguntas de búsqueda

Lo que distingue la revisión sistemática de otro tipo de estudios informales, es que la revisión sistemática debe definir las preguntas de investigación antes de iniciar el estudio. En el presente estudio dado la necesidad de un estudio que profundice el conocimiento en una arquitectura de servicios web semánticos basada en aspectos de la composición y seguridad. Se tiene las siguientes preguntas de investigación:

- RQ1: What are the modules of architecture of the composite semantic web service?
- RQ2: How to use ontologies in architecture of the composite semantic web service?
- RQ3: How to support security capabilities in architecture of the composite semantic web service?
- RQ4: What are composition techniques used in architecture of the composite semantic web service?
- RQ5: How to use artificial intelligence in architecture of the composite semantic web service?

## 4.2 Estrategia de la búsqueda

Definir la estrategia de búsqueda es otra de las etapas importantes dentro de una revisión sistemática, en esta etapa tiene como objetivo identificar toda la información relevante para la investigación, documentando los resultados obtenidos de manera que se pueda evaluar el rigor con que fue hecho el estudio y que el proceso pueda ser repetido.

### 4.2.1 Derivando los términos de búsqueda

En este punto se obtiene los términos de búsqueda, extrayéndolos de las preguntas de investigación. En la tabla 4.1 se muestran las palabras por cada pregunta.

<b>RQ1</b>	module, architecture , semantic web service , composite.
<b>RQ2</b>	ontology, architecture , semantic web service , composite.
<b>RQ3</b>	capability, architecture , semantic web service , composite , security
<b>RQ4</b>	techniques, architecture , semantic web service , composite.
<b>RQ5</b>	artificial intelligence , architecture , semantic web service , composite

Tabla 4.1 Palabras clave identificadas de las preguntas de búsqueda.

### 4.2.2 Definiendo las cadenas de búsqueda

Se hizo búsquedas preliminares en Google Scholar para determinar si existen revisiones sistemáticas existentes. La cadena de búsqueda definida fue “Systematic Review semantic web service composition architecture”

La búsqueda fue llevada a cabo en mayo del 2017 y no se encontró revisiones sistemáticas sobre el tema. Luego en abril del 2018 se efectuó de nuevo la búsqueda no encontrándose ninguna revisión sistemática sobre el tema.

Luego se hizo una búsqueda en la base de datos SCOPUS con la finalidad de identificar términos relacionados que permitan completar y enriquecer las cadenas de búsqueda para no omitir ninguna investigación relevante, siendo la cadena de búsqueda la siguiente:

**( TITLE-ABS-KEY ( architecture ) AND TITLE-ABS-KEY ( "semantic web service" ) AND TITLE-ABS-KEY ( composition ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) )**

La búsqueda fue llevada a cabo en abril del 2018 y se encontraron 218 estudios, se revisaron los títulos, los términos claves y los resúmenes de todos los artículos encontrados. Como resultado de esa lectura se encontraron dos conceptos recurrentes tales como la ontología y la inteligencia artificial. Debido a la importancia de esos términos, se le agregó a las preguntas de investigación del punto 4.1.

Por lo tanto, la cadena de búsqueda por cada pregunta de investigación resulta de la combinación de los sinónimos derivados de los tesauros del diccionario de Oxford de los términos de búsqueda definidos en el punto 4.2.1 y los nuevos términos obtenidos en la búsqueda de la base de datos SCOPUS. La tabla 4.2 muestra los términos a utilizar en las cadenas de búsqueda.

<b>architecture</b>	<b>system architecture</b>
<b>module</b>	layer , component
<b>composite</b>	composition, composing , choreography, orchestration
<b>capability</b>	feature , requirement , requisite
<b>security</b>	secure, authentication, authorization, access control, trust, policy
<b>semantic web service</b>	semantic service
<b>techniques</b>	mechanism , method, algorithm

Tabla 4.2 Sinónimos y términos relacionados a las palabras claves identificados en la tabla 4.1

## 4.2.3 Recursos de búsqueda

Las Bases de datos utilizadas para esta investigación son las siguientes:

- SCOPUS
- IEEE
- ACM DL

## 4.2.4 Definiendo cadenas de búsqueda por cada motor de búsqueda

Las cadenas de búsqueda para responder cada pregunta de investigación de acuerdo a la base de datos SCOPUS, es la mostrada en la tabla 4.3

<b>RQ1</b>	<b>TITLE-ABS-KEY ( module OR layer OR component )</b> <b>AND TITLE-ABS-KEY ( architecture OR framework)</b> <b>AND TITLE-ABS-KEY ( "semantic web service*" OR "semantic service*")</b> <b>AND TITLE-ABS-KEY ( composite OR composition OR composing OR choreography OR orchestration )</b>
<b>RQ2</b>	TITLE-ABS-KEY ( ontology ) AND TITLE-ABS-KEY ( architecture OR framework ) AND TITLE-ABS-KEY ( "semantic web service*" OR "semantic service*" ) AND TITLE-ABS-KEY ( composite OR composition OR composing OR choreography OR orchestration )
<b>RQ3</b>	TITLE-ABS-KEY ( requirement OR requisite OR capability OR feature ) AND TITLE-ABS-KEY ( architecture OR framework ) AND TITLE-ABS-KEY ( "semantic web service*" OR "semantic service*" ) AND TITLE-ABS-KEY ( composite OR composition OR composing OR choreography OR orchestration ) AND TITLE-ABS-KEY ( security OR secure OR authentication OR authorization OR "access control" OR trust OR policy )
<b>RQ4</b>	TITLE-ABS-KEY ( techniques OR mechanism OR method OR algorithm ) AND TITLE-ABS-KEY ( architecture OR framework ) AND TITLE-ABS-KEY ( "semantic web service*" OR "semantic service*" ) AND TITLE-ABS-KEY ( composite OR composition OR composing OR choreography OR orchestration )

<b>RQ5</b>	TITLE-ABS-KEY ("artificial intelligence") AND TITLE-ABS-KEY ( architecture OR framework) AND TITLE-ABS-KEY ( "semantic web service*" OR "semantic service*") AND TITLE-ABS-KEY ( composite OR composition OR composing OR choreography OR orchestration )
------------	--

Tabla 4.3 Cadena de búsqueda adaptada a la base de datos SCOPUS

La cadena de búsqueda para responder cada pregunta de investigación de acuerdo a la base de datos IEEE, es la mostrada en la tabla 4.4

<b>RQ1</b>	<b>((((module OR layer OR component ) AND (architecture OR framework) AND ("semantic web service*" OR "semantic service*") AND (composite OR composition OR composing OR choreography OR orchestration))))</b>
<b>RQ2</b>	<b>(((((ontology) AND (architecture OR framework) AND ("semantic web service*" OR "semantic service*") AND (composite OR composition OR composing OR choreography OR orchestration) )))</b>
<b>RQ3</b>	<b>((requirement OR requisite OR capability OR feature) AND (architecture OR framework) AND ("semantic web service*" OR "semantic service*") AND (composit* OR composing OR choreography OR orchestration) AND ( secur* OR auth* OR "access control" OR trust OR policy))</b>
<b>RQ4</b>	<b>(((((techniques OR mechanism OR method) AND (architecture OR framework) AND ("semantic web service*" OR "semantic service*") AND (composite OR composition OR composing OR choreography OR orchestration) )))</b>
<b>RQ5</b>	<b>(((((("artificial intelligence") AND (architecture OR framework) AND ("semantic web service*" OR "semantic service*") AND (composite OR composition OR composing OR choreography OR orchestration))))</b>

Tabla 4.4 Cadena de búsqueda adaptada a la base de datos IEEE

La cadena de búsqueda para responder cada pregunta de investigación de acuerdo a la base de datos ACM DL, es la mostrada en la tabla 4.5

<b>RQ1</b>	<b>(module layer component)</b> <b>+(architecture framework)</b> <b>+("semantic web service*" "semantic service*")</b> <b>+(composite OR composition OR composing OR choreography OR orchestration)</b>
------------	--

<b>RQ2</b>	(ontology) +(architecture framework) +("semantic web service*" "semantic service*") +(composite OR composition OR composing OR choreography OR orchestration)
<b>RQ3</b>	(requirement requisite capability feature) +(architecture framework) +("semantic web service*" "semantic service*") +(composite OR composition OR composing OR choreography OR orchestration) +(security secure authentication authorization "access control" trust policy)
<b>RQ4</b>	(techniques mechanism method algorithm) +(architecture framework) +("semantic web service*" "semantic service*") +(composite OR composition OR composing OR choreography OR orchestration)
<b>RQ5</b>	("artificial intelligence") +(architecture framework) +("semantic web service*" "semantic service*") +(composite OR composition OR composing OR choreography OR orchestration)

Tabla 4.5 Cadena de búsqueda adaptada a la base de datos ACM DL

## 4.2.5 Proceso de búsqueda

Este proceso consta de tres etapas:

La primera etapa consiste en dos búsquedas preliminares, donde se tiene por objetivo determinar la existencia de una revisión sistemática sobre el tema que aborda la presente investigación, y la búsqueda de nuevos términos que permitan agregar nuevas preguntas de investigación o enriquecer las preguntas existentes.

La segunda etapa consiste en seleccionar los artículos encontrados luego de ejecutar la cadena de búsqueda en los recursos descritos en el punto 4.2.4, por tal motivo se hace uso de los criterios de inclusión y exclusión presentados en el punto 4.3.1, los cuales son aplicados al título, resumen y

palabras clave de cada artículo . En caso de no tener la certeza del contenido del artículo solo con leer el resumen del artículo, se procederá a leer todo el artículo.

La tercera etapa consiste en obtener los artículos finales que van a responder las preguntas de investigación, en esta etapa se utilizan los criterios de aseguramiento de calidad, los criterios de inclusión y exclusión y la lista de verificación de aseguramiento de calidad, siendo necesario leer todo el artículo enfocándose en la introducción y las conclusiones.

## 4.2.6 Documentando el Proceso de búsqueda

Documentar el proceso de búsqueda, es muy importante ya que permite que un investigador pueda evaluar el rigor de la revisión sistemática y también pueda replicar el proceso. Para las búsquedas en base de datos electrónicas se siguieron los procedimientos de búsqueda dados por Kitchenham [16] , se documentó la siguiente información de los recursos utilizados para la revisión sistemática:

- Nombre de la base de datos.
- Estrategia de búsqueda para la base de datos.
- Fecha de búsqueda.
- Años cubiertos por la búsqueda.
- Número de duplicados.
- Número de artículos rechazados.
- Número de artículos seleccionados.

## 4.3 Criterio de la evaluación de la calidad del estudio

Luego de definir las preguntas y la estrategia de búsqueda, es necesario identificar que estudios primarios serán incluidos o serán excluidos de la revisión sistemática, debido a que una revisión sistemática se caracteriza por hacer uso de un proceso riguroso de búsqueda a diferencia de otros tipos de estudios tradicionales, se definen a continuación los criterios de inclusión y exclusión así como la lista de verificación de calidad para establecer una búsqueda no sesgada.



### 4.3.1 Criterio de inclusión y exclusión para la selección del estudio

Los criterios de inclusión y exclusión son utilizados para evaluar la calidad y para seleccionar los estudios primarios. Estos criterios se basan en las preguntas de investigación y aunque son definidos antes de iniciar la búsqueda, podrían ser redefinidos durante el proceso de búsqueda en caso de requerirse mejoras.

Todo estudio primario para ser incluido debe satisfacer los cuatro primeros criterios de inclusión, el criterio 5 es para seleccionar los estudios relacionados con la pregunta 2, el criterio 6 es para seleccionar los estudios relacionados con la pregunta 3, el criterio 7 es para seleccionar los estudios relacionados con la pregunta 4 y el criterio 8 es para seleccionar los estudios relacionados con la pregunta 5. Los criterios de inclusión son los siguientes:

1. Estudios en inglés.
2. Estudios cuyas propuestas presentan una arquitectura que permiten soportar la composición de servicios web.
3. Estudios que explican con detalle los componentes de la arquitectura.
4. Estudios que tratan de servicios web semánticos
5. Estudios que definen una arquitectura basadas en el uso de ontologías para la composición de los servicios web semánticos
6. Estudios que definen una arquitectura de servicios web semánticos considerando aspectos de seguridad en la composición de los servicios web.
7. Estudios que explican en forma detallada el criterio utilizado para componer un servicio web semántico.
8. Estudios que definen el rol de la inteligencia artificial en la composición de servicios web semánticos.

A continuación se define estos criterios para identificar los artículos que no proporcionan información relevante para la revisión sistemática.

1. Estudios escritos en otros idiomas diferentes al inglés.
2. Estudios que no proponen ningún tipo de arquitectura que soportan la composición de servicios web semánticos.
3. Estudios que no explican los componentes de la arquitectura.
4. Estudios basados en servicios web sintácticos.
5. Estudios cuya arquitectura no considera el uso de ontologías.
6. Estudios que presentan la composición de servicios web semánticos y la seguridad en forma separada.
7. Estudios que no se basan en ningún criterio para componer un servicio web semántico.
8. Estudios que no explica el uso de la inteligencia artificial en la composición de servicios web semánticos
9. Estudios que son libros, capítulos de libros, actas de conferencias o estudios secundarios o terciarios.

### 4.3.2 Lista de aseguramiento de calidad

La lista de preguntas para evaluar la calidad y credibilidad están basadas en la lista de preguntas presentadas por Kitchenham [16], combinadas con preguntas propias de la investigación. Estas son las siguientes:

1. ¿Se indica claramente el objetivo del estudio?
2. ¿Se explica en forma clara la arquitectura?
3. ¿Se explica en forma clara cada componente de la arquitectura?
4. ¿Se explica en forma clara la composición de los servicios web semánticos en la arquitectura?
5. ¿Las conclusiones son claras y coherentes?

Cada pregunta tiene tres opciones de respuesta que siguen la siguiente escala: sí (1), no (0), parcialmente (0.5). Estas preguntas de aseguramiento de calidad, serán aplicadas luego de obtener los resultados en la primera búsqueda, para obtener los artículos finales que van a responder a las preguntas de investigación. Como criterio cada artículo debe tener un valor mayor a 3 para ser considerado.

### 4.3.3 Estrategia de extracción de Datos

Una vez definido los criterios de inclusión y exclusión así como la lista de aseguramiento de calidad, se debe definir los mecanismos necesarios para recolectar la información de los estudios primarios que respondan satisfactoriamente a las preguntas de investigación. En esta investigación se utilizan el formulario de extracción de datos y el procedimiento de extracción de data, lo cuales se detallan a continuación.

### 4.3.4 Formulario de Extracción de Datos

El formulario de extracción de data es un instrumento que sirve para extraer la información de todos los artículos relevantes que respondan satisfactoriamente las preguntas de investigación y que sobrepasen los criterios de calidad. Este formulario debe ser definido en el protocolo de revisión antes de iniciar la búsqueda y no está exento a mejoras en caso de encontrar dificultades en el registro de información. Para la presente revisión sistemática se utiliza el formulario de extracción de data mostrado en la tabla 4.6. El formulario está compuesto por una sección de información estándar, por ejemplo: Autores, título etc., y por otra sección con referencia a las preguntas de investigación.

Campo	Descripción	RQ
ID	Número secuencial	General
Fecha de Extracción		General
Autores		General

<b>Título</b>		General
<b>Tipo de Estudio</b>	Si es un artículo de revista o un artículo de conferencia	General
<b>Nombre del recurso de búsqueda</b>	Nombre del recurso donde fue encontrado el estudio	General
<b>Año de Publicación</b>		General
<b>Institución</b>	Institución que representan los investigadores	General
<b>País</b>		General
<b>Problema a resolver</b>	Breve descripción del principal problema que la arquitectura pretende resolver	General
<b>Tipo de arquitectura</b>	Si la arquitectura es conceptual o concreta	General
<b>Campo de aplicación</b>	En qué ámbito la arquitectura va ser aplicada	General
<b>Módulos de la arquitectura</b>	Lista de los módulos que tiene la arquitectura	RQ1
<b>Patrón de arquitectura</b>	Lista de patrones sobre la que se diseña la arquitectura	RQ1
<b>Nivel de automatización</b>	Si la arquitectura soporta una composición de servicios de tipo manual , semiautomático o automatizado	RQ1
<b>Lenguaje</b>	Lista de Lenguajes de Ontologías empleadas en la arquitectura	RQ2
<b>Fase</b>	Etapa de la arquitectura en el cual interviene la ontología	RQ2

<b>Propósito</b>	Objetivo de la Ontología Seleccionada	RQ2
<b>Etapa</b>	Etapa de la arquitectura en la cual son tomados en cuenta los requerimientos de seguridad	RQ3
<b>Componente</b>	Componente de la arquitectura que soporta la seguridad	RQ3
<b>Naturaleza</b>	Campo de aplicación al que pertenece el Algoritmo y/Criterio utilizado	RQ4
<b>Algoritmo y/técnica</b>	Lista de Algoritmo y/o técnica utilizada en la arquitectura	RQ4
<b>Rol</b>	Función de la Inteligencia artificial en la arquitectura	RQ5
<b>Etapa</b>	Etapa de la arquitectura en el cual interviene la inteligencia artificial	RQ5

Tabla 4.6 Formulario de Extracción de data

### 4.3.5 Procedimiento de Extracción de Datos

Como procedimiento de extracción de data, se hará uso del software especializado en revisiones sistemáticas denominado: StArt .Esta herramienta será utilizada en la búsqueda primaria en una primera etapa y luego en una búsqueda secundaria. Esta herramienta permite tener un consolidado final de artículos mediante el uso del protocolo de revisión, el cual es soportado por la herramienta.

# Capítulo 5.

## Ejecución

Una vez definido el protocolo de revisión, se ejecuta el proceso de búsqueda, mostrado en forma resumida en el punto 4.2.5, en este punto es donde se evalúa el protocolo de revisión de modo que si en caso se identificara la necesidad de efectuar ajustes se debe de modificar el protocolo de revisión y ejecutar de nuevo el proceso de búsqueda.

### 5.1 Búsqueda en las Base de datos

Las búsquedas se ejecutaron desde el 11 de abril de 2018. En la tabla 5.1 se presenta datos importantes tales como la fecha de búsqueda y el intervalo de tiempo al que pertenece cada artículo obtenido después de la ejecución

Recurso de Búsqueda	Estrategia de búsqueda	Fechas de la Búsqueda	Años cubiertos
SCOPUS	Cadena de búsqueda en 4.2.4	RQ1: 11/04/2018 RQ2: 11/04/2018 RQ3: 11/04/2018 RQ4: 11/04/2018 RQ5: 11/04/2018	RQ1: 2003-2017 RQ2: 2003-2018 RQ3: 2003-2017 RQ4: 2003-2018 RQ5: 2004-2016
IEEE	Cadena de búsqueda en 4.2.4	RQ1: 12/04/2018 RQ2: 12/04/2018 RQ3: 12/04/2018 RQ4: 12/04/2018	RQ1: 2004-2015 RQ2: 2002-2016 RQ3: 2004-2013 RQ4: 2004-2016

		RQ5: 12/04/2018	RQ5: 2005-2016
ACM DL	Cadena de búsqueda en 4.2.4	RQ1: 13/04/2018 RQ2: 13/04/2018 RQ3: 13/04/2018 RQ4: 13/04/2018 RQ5: 13/04/2018	RQ1: 2002-2010 RQ2: 2002-2010 RQ3: 2003-2011 RQ4: 2002-2010 RQ5: 2002-2010

Tabla 5.1 Registro de la ejecución de las búsquedas

La tabla 5.2 muestra el número de artículos arrojados por cada cadena en búsqueda en cada base de datos. Se puede ver en forma preliminar que la mayor cantidad de artículos provienen de la base de datos SCOPUS, y la pregunta RQ4 tiene la mayor cantidad de artículos disponibles de un total de 793 artículos encontrados.

Cadena de Búsqueda	SCOPUS	IEEE	ACM DL	Total
RQ1	66	21	31	124
RQ2	158	81	31	270
RQ3	12	8	6	26
RQ4	171	71	31	273
RQ5	19	56	31	106
Total	426	243	130	<b>793</b>

Tabla 5.2 Número de búsqueda por cada cadena de búsqueda y recurso de búsqueda

Los artículos mostrados en la tabla 5.2, no son los artículos finales, antes se tiene que pasar por un conjunto de filtros definidos en el protocolo de revisión y también eliminar los artículos duplicados. Las fases del proceso de búsqueda se muestran en la figura 5.1

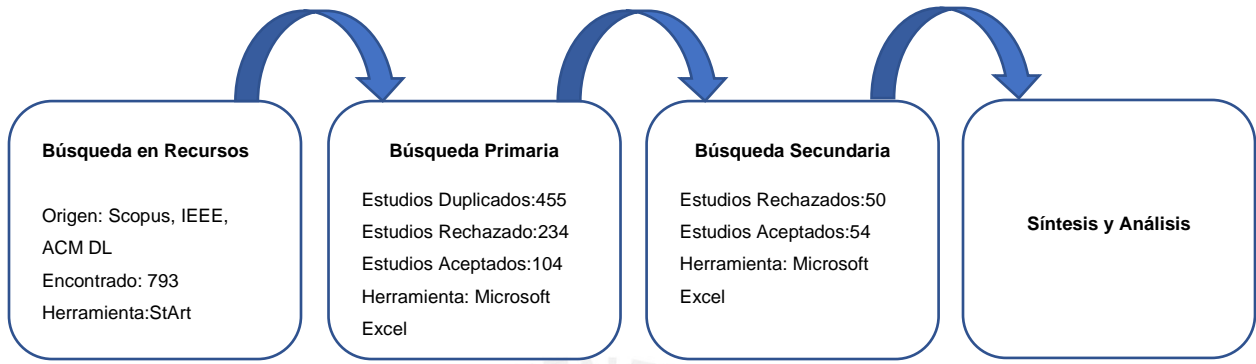


Figura 5.1 Proceso de búsqueda

## 5.2 Búsqueda primaria

En esta fase, debido a la gran cantidad de artículos encontrados en las base de datos, los cuales son mostrados en la tabla 5.2, fue necesario el empleo de una herramienta que permitiera un mejor control en la revisión de los artículos, se investigó varias herramientas, sin embargo debido a su facilidad en el uso, se eligió la herramienta StArt. Esta herramienta requería el ingreso de todo el protocolo de revisión, además también requería que todos los resultados de las búsquedas fueran importadas en formato BibTex.

En este punto es necesario resaltar que cuando se importó la búsqueda de la pregunta 1 efectuada en la base de datos IEEE se generó 6 artículos sobrantes, estos artículos no se eliminaron en la herramienta StArt, solo fueron tratados como rechazado y duplicados, asimismo cuando se utilizó la funcionalidad de la herramienta en la detección de duplicados, se comprobó posteriormente que existía duplicados que la herramienta no lograba identificar, debido a que existía artículos que tenían una leve diferencia en los títulos de los artículos en cada base de datos y también una diferencia en el orden de los autores, sin embargo eran similares en el resumen y el contenido. Dado la situación anterior se hizo necesario una revisión manual analizando los títulos, los autores y los resúmenes, utilizando el Microsoft Excel y la técnica de las tablas dinámicas



Cuenta de ID Paper	Status/Selection	ACCEPTED	DUPLICATED	REJECTED	Total general
BD					
ACM		6	111	13	130
IEEE		41	123	73	237
SCOPUS		57	221	148	426
Total general		104	455	234	793

Figura 5.2 Tablas Dinámicas.

Luego se actualizó el campo de los duplicados en la herramienta StArt, con los resultados obtenidos de la revisión anterior, donde se encontró 459 duplicados. Una vez identificados todos los duplicados, se procedió a la revisión de los artículos no duplicados, utilizando los criterios de inclusión y de Exclusión

5673 - Towards semantic services for sensor-rich information systems

Study Data Selection Data Data Extraction Form Similar Studies

URL: \_\_\_\_\_

Volume: \_\_\_\_\_ Pages: 967-974 Vol. 2 ISSN / ISBN: \_\_\_\_\_

DOI: 10.1109/ICBN.2005.1589709 Importation date: 04/29/2018

**Study selection criteria**

- (I) The study must be written in English
- (I) The studies which explain an architectures considering security
- (I) The studies which explain in great detail all criterion used to compose
- (I) The studies which explain the rol of the intelligence artificial
- (E) The study does not written in English
- (E) The study which does not propose an architecture which support
- (E) Those that don't include an explanation of the modules that compose
- (E) The studies explains sintactic web services
- (E) The studies which explain architectures which do not use ontologies
- (E) The studies which do not consider security within of an architecture
- (E) The studies which do not explain any criterion used to compose
- (E) The studies which do not explain the rol of the intelligence artificial
- (E) Those that are either books, conference proceedings, or security

**Selected criteria**

- (I) The study proposes an architecture which support semantic v
- (I) The study explains in greater detail the modules the archite
- (I) The studies explains semantic web services
- (I) The studies which describe architectures based on ontologies

Status: Accepted Search session: SEARCH1 *This paper is in Extraction step*

Reading Priority: Low Score: 36

Full text

save & previous save & next

previous next

Save Cancel

Figura 5.3 Herramienta StArt Criterios de Inclusión y de Exclusión

Luego de una profunda revisión, que duró aproximadamente un mes, se obtuvo los resultados mostrados en la tabla 5.3 y en la figura 4, siendo la base de datos Scopus la que tiene mayor cantidad de artículos aceptados seguida ligeramente por la base de datos IEEE.

Base de datos	Duplicado	Rechazado	Aceptado	Total
IEEE	123	73	<b>41</b>	237
Scopus	221	148	<b>57</b>	426
ACM	111	13	<b>6</b>	130
<b>Total</b>	<b>455</b>	<b>234</b>	<b>104</b>	<b>793</b>

Tabla 5.3 Resumen de la primera búsqueda

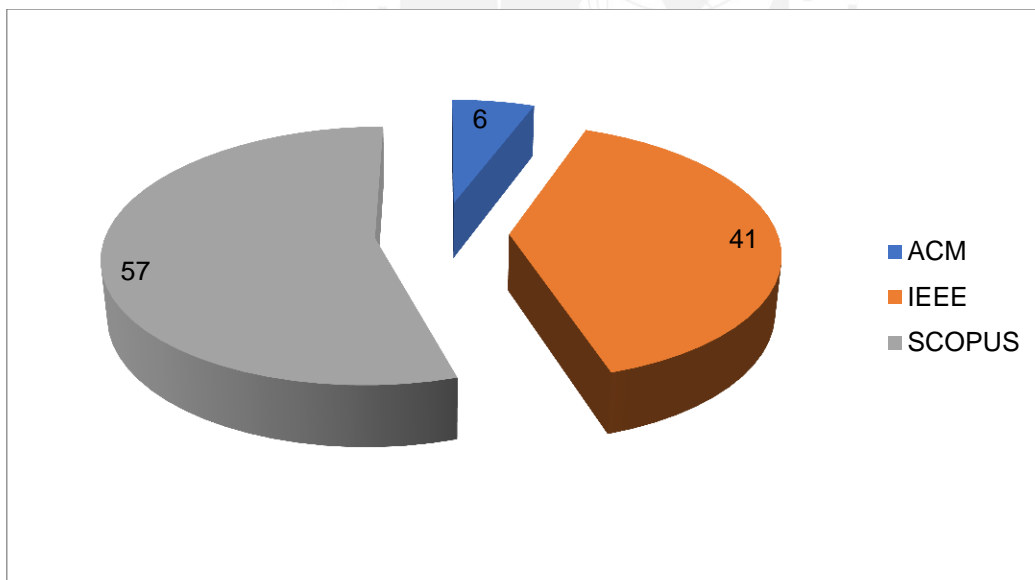


Figura 5.4 Gráfico estadístico

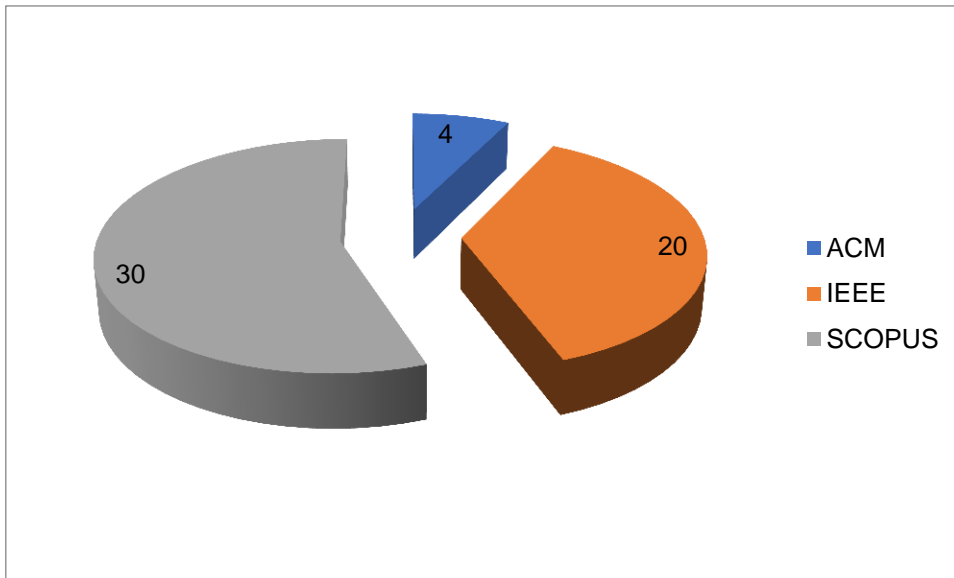
## 5.3 Búsqueda secundaria

En esta fase los estudios encontrados en la fase de la búsqueda primaria son nuevamente filtrados utilizando los criterios de inclusión y exclusión del punto 4.3.1 combinados con los criterios de calidad del punto 4.3.2 .La herramienta StArt ya no fue utilizada en la búsqueda secundaria excepto en la exportación de los estudios, debido a que la cantidad de artículos a analizar se redujo considerablemente. En su lugar se empleó el Microsoft Excel para el empleo de sus tablas dinámicas y de sus gráficos estadísticos.

Esta fase tomo un considerable tiempo debido a la búsqueda del artículo completo y a la necesidad leer la totalidad del artículo muchos de los cuales fue necesario leerlos en repetidas veces, debido a la gran complejidad técnica en los conceptos que utilizaba. Los resultados de esta fase son mostrados en la tabla 5.4

Base de datos	Rechazado	No Disponible	Aceptado
IEEE	21	-	20
Scopus	27	-	30
ACM	2	-	4
Total	50	-	54

Tabla 5.4 Resumen de la búsqueda secundaria



**Figura 5.5 Gráfico estadístico**

Otra forma de como clasificar los resultados de la búsqueda secundaria, teniendo en cuenta, el artículo, el año y la pregunta de búsqueda por el cual fueron seleccionados los artículos son mostrados en la tabla 5.5

Artículo	Pregunta de Búsqueda	Base de Datos	Año
[18]	RQ1 RQ2 RQ4 RQ5	IEEE	2008
[19]	RQ1 RQ2 RQ4 RQ5	IEEE	2009
[20]	RQ1 RQ2 RQ4 RQ5	IEEE	2010
[21]	RQ1 RQ2 RQ3 RQ4	IEEE	2009
[22]	RQ1 RQ2 RQ4	IEEE	2005
[23]	RQ1 RQ2 RQ4	IEEE	2004
[24]	RQ1 RQ2 RQ3 RQ4	IEEE	2010
[25]	RQ1 RQ2 RQ4	IEEE	2008
[26]	RQ1 RQ2 RQ4 RQ5	IEEE	2005

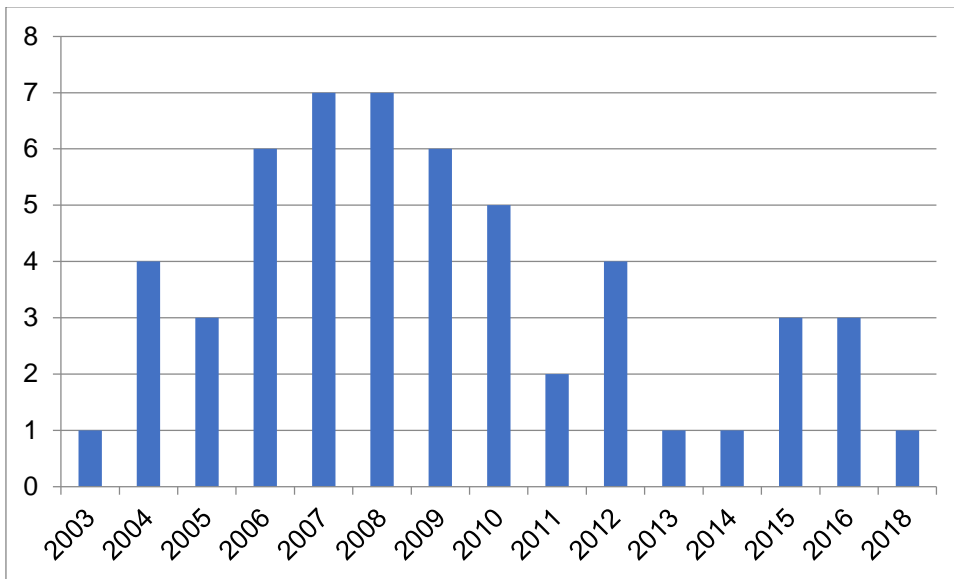
[27]	RQ1 RQ2 RQ4 RQ5	IEEE	2010
[28]	RQ1 RQ2 RQ4	IEEE	2013
[29]	RQ1 RQ2 RQ4	IEEE	2008
[30]	RQ1 RQ2 RQ4	IEEE	2007
[31]	RQ1 RQ2 RQ4	IEEE	2005
[32]	RQ1 RQ2 RQ4	IEEE	2007
[33]	RQ1 RQ2 RQ4 RQ5	IEEE	2009
[34]	RQ1 RQ2 RQ4	IEEE	2007
[35]	RQ1 RQ2 RQ4	IEEE	2009
[36]	RQ1 RQ2 RQ4 RQ5	IEEE	2008
[37]	RQ1 RQ2 RQ4 RQ5	IEEE	2006
[38]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2016
[39]	RQ1 RQ2 RQ3 RQ4 RQ5	SCOPUS	2014
[40]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2012
[41]	RQ1 RQ2 RQ4	SCOPUS	2008
[42]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2007
[43]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2018
[44]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2016
[45]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2015
[46]	RQ1 RQ2 RQ4	SCOPUS	2012
[47]	RQ1 RQ2 RQ4	SCOPUS	2010
[48]	RQ1 RQ2 RQ4	SCOPUS	2010
[49]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2009
[50]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2009

[51]	RQ1 RQ2 RQ4	SCOPUS	2008
[52]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2008
[53]	RQ1 RQ2 RQ4	SCOPUS	2007
[54]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2007
[55]	RQ1 RQ2 RQ3 RQ4 RQ5	SCOPUS	2007
[56]	RQ1 RQ2 RQ4	SCOPUS	2004
[57]	RQ1 RQ2 RQ4	SCOPUS	2004
[58]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2015
[59]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2015
[60]	RQ1 RQ2 RQ3 RQ4 RQ5	SCOPUS	2012
[61]	RQ1 RQ2 RQ4 RQ5	SCOPUS	2012
[62]	RQ1 RQ2 RQ4	SCOPUS	2011
[63]	RQ1 RQ2 RQ4	SCOPUS	2006
[64]	RQ1 RQ2 RQ4	SCOPUS	2006
[65]	RQ1 RQ2 RQ3 RQ4 RQ5	SCOPUS	2006
[66]	RQ1 RQ2 RQ3 RQ4 RQ5	SCOPUS	2006
[67]	RQ1 RQ2 RQ3 RQ4	SCOPUS	2004
[68]	RQ1 RQ2 RQ4	ACM	2016
[69]	RQ1 RQ2 RQ3 RQ4	ACM	2003
[70]	RQ1 RQ2 RQ4	ACM	2011
[71]	RQ1 RQ2 RQ4	ACM	2006

**Tabla 5.5 Estudios seleccionados al final de la segunda búsqueda**

En la tabla 5.5 se puede identificar que los estudios seleccionados cubren los años desde el 2003 hasta el 2018, además utilizando los resultados presentados en la tabla 5.4 se puede identificar que la mayoría de estos estudios provienen de la base de datos SCOPUS.

En la figura 5.6 se muestra en forma visual la información presentada en la tabla 5.5 , en esta figura se puede identificar que los años 2007 y 2008 se encuentran la mayor cantidad de artículos seleccionados, cada uno con 7 artículos respectivamente , esto se debe principalmente a que los artículos ubicados dentro de esos años, son los que satisfacían los criterios de inclusión y exclusión y la lista de aseguramiento de calidad , siendo los criterios de inclusión más exigentes: a) Estudios cuyas propuestas presentan una arquitectura que permiten soportar la composición de servicios web y b) Los estudios que explican con detalle los componentes de la arquitectura.



**Figura 5.6 Artículos clasificados por año**

# Capítulo 6.

## Síntesis y Análisis de Datos

En las siguientes secciones con la información obtenida de los estudios seleccionados mostrados en la tabla 5.5, se procederá a resolver los hechos generales y las preguntas de investigación de acuerdo al formulario de extracción de datos.

### 6.1 Hechos generales de los estudios seleccionados

#### 6.1.1 Resultados de la listas de verificación de aseguramiento de calidad

Los resultados de la lista de verificación de la evaluación de calidad se presentan en la tabla 6.1. Estos muestran que el objetivo de cada estudio se encontró claramente establecido. La segunda pregunta trata de identificar si la arquitectura fue explicada claramente en el estudio, es decir explicar en forma resumida cómo trabajan conjuntamente cada componente para lograr la composición de los servicios web semánticos. La tercera pregunta intenta verificar si cada componente de la arquitectura fue explicada en forma detallada y clara. La cuarta pregunta intenta de verificar si el criterio utilizado en la composición de servicios web semánticos es presentado en forma clara y detallada. La quinta pregunta se verifica que las conclusiones estuvieron bien definidas, detalladas y alineadas al objetivo de la investigación, evitando declaraciones simples u obvias que quitarían el valor académico al artículo seleccionado.



ID	Pregunta de calidad	Sí	No	Parcialmente
1	¿Se indica claramente el objetivo del estudio?	54	0	0
2	¿Se explica en forma clara la arquitectura?	54	0	0
3	¿Se explica en forma clara cada componente de la arquitectura?	52	0	2
4	¿Se explica en forma clara la composición de los servicios web semánticos en la arquitectura?	52	0	2
5	¿Las conclusiones son claras y coherentes?	54	0	0

Tabla 6.1 Resultado de la lista de verificación de aseguramiento de calidad

## 6.1.2 Instituciones y Países de la investigación

En la tabla 6.2 se puede observar lo siguiente, de acuerdo a una clasificación regional, se observa que la mayoría de los estudios fueron efectuados en Europa, sin embargo de acuerdo a una clasificación por país, se observa que la mayoría de estudios fueron dados en China. Además cabe mencionar que en este nivel de investigación se aprecia un trabajo colaborativo donde trabajan en forma conjunta varias instituciones de diferentes países.

País(es)	Cantidad
Alemania	3
Alemania, Austria	1
Argelia	3
Brasil	2
China	9
Colombia	2
Corea del Sur	1
España	2
Finlandia, España	2
Francia	2
Holanda	1
Holanda, Reino Unido	1
India	3
Irán	1
Irlanda	1
Italia	2
Italia, España, Suiza, Finlandia, Alemania, Portugal	1
Malasia	1
Marruecos	2
Noruega	1

Noruega, Suecia	2
Reino Unido , China	1
Singapur	2
Suecia , Italia , Noruega	1
Taiwán	1
Turquía	1
Estados Unidos	5
<b>Total general</b>	<b>54</b>

Tabla 6.2 Cuadro resumen de estudios por país

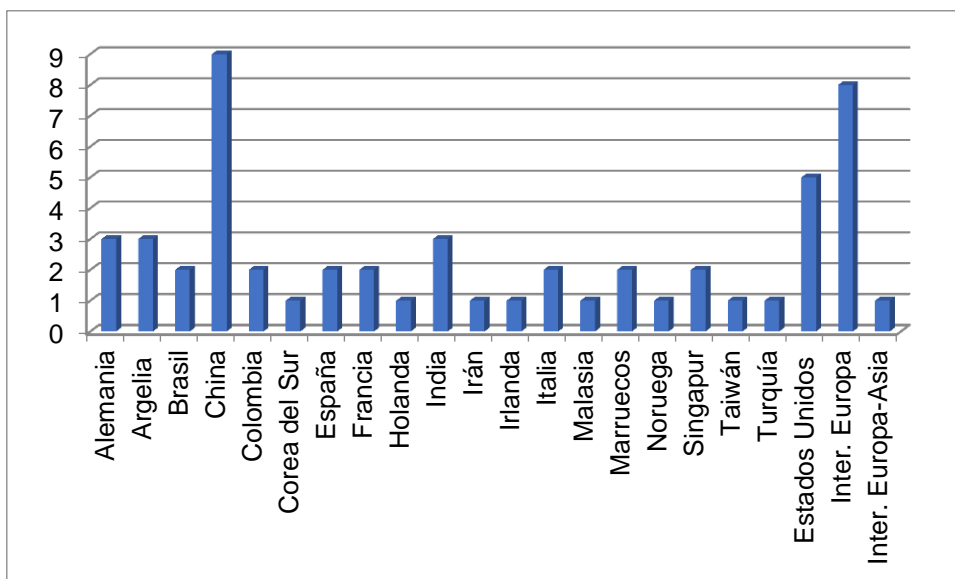


Figura 6.1 Estudios por país

Artículo	País
[18]	Malasia
[19]	Suecia , Italia , Noruega
[20]	China
[21]	China
[22]	China
[23]	Noruega, Suecia
[24]	Irán
[25]	China
[26]	Reino Unido , China
[27]	China
[28]	Argelia
[29]	Alemania
[30]	Alemania

[31]	Singapur
[32]	Corea del Sur
[33]	Colombia
[34]	Singapur
[35]	India
[36]	Colombia
[37]	Francia
[38]	Finlandia, España
[39]	Brasil
[40]	Argelia
[41]	Irlanda
[42]	Alemania
[43]	India
[44]	Estados Unidos
[45]	Marruecos
[46]	Taiwán
[47]	Argelia
[48]	Estados Unidos
[49]	China
[50]	China
[51]	Italia
[52]	Estados Unidos
[53]	España
[54]	España
[55]	China
[56]	Usa
[57]	Brasil
[58]	Marruecos
[59]	Finlandia, España
[60]	China
[61]	Alemania, Austria
[62]	India
[63]	Francia
[64]	Italia
[65]	Italia, España, Suiza, Finlandia, Alemania, Portugal
[66]	Noruega ,Suecia
[67]	Noruega
[68]	Holanda, Reino Unido
[69]	Estados Unidos
[70]	Holanda
[71]	Turquía

**Tabla 6.3 Cuadro detallado de estudios por país**

### 6.1.3 Campos de aplicación y problemas que resuelven

De la tabla 6.4 se puede observar que se tienen dos tipos de arquitecturas de acuerdo al campo de aplicación, las cuales son denominadas arquitectura de propósito general, y arquitectura de propósito específico. Siendo la mayoría de investigaciones formuladas bajo un propósito general es decir puede ser aplicada en cualquier campo. En cuanto a las arquitecturas que tienen un campo de aplicación específico, se puede observar que las arquitecturas en los sistemas de fabricación y en las redes son los más investigados.

En cuanto a los problemas a ser resueltos por la arquitectura, la mayoría tiende a proponer una composición automática y dinámica, es decir la composición de servicios web semánticos se desarrollan con una nula participación de un usuario experto y además son reactivos a los cambios generando una nueva composición en caso que un componente ha sido removido o presento fallas en su funcionamiento.

Campo de aplicación	Estudios	Conteo	%
Propósito General	[18],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[32],[33],[35],[36],[37],[39],[40],[41],[42],[43],[44],[45],[46],[47],[49],[50],[51],[53],[54],[55],[56],[60],[62],[63],[64],[66],[67],[68],[69],[70],[71]	43	79.63
Pequeña y Mediana Empresa	[61]	1	1.85
Sistemas Multi-Robots (MRS)	[19]	1	1.85
Manufactura	[31],[38],[59]	3	5.56
Empresas virtuales	[34]	1	1.85
Batalla	[48]	1	1.85
Gestión de Modelos	[52]	1	1.85
Comercio electrónico empresa a empresa.	[57]	1	1.85
Gobierno electrónico	[58]	1	1.85
Salud	[65]	1	1.85

**Tabla 6.4 Campo de aplicación**

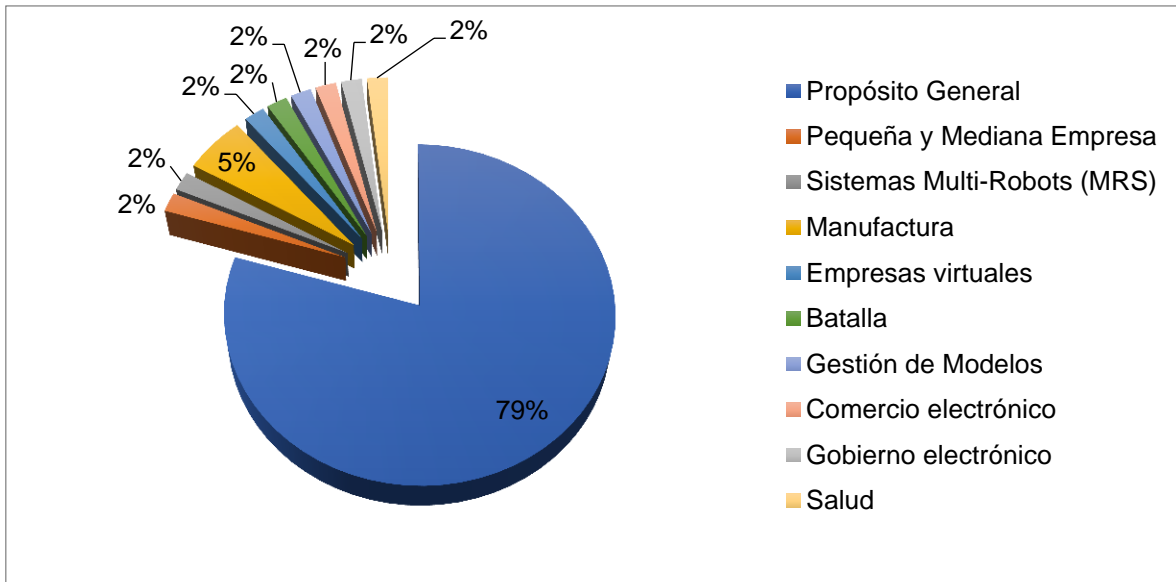


Figura 6.2 Campo de aplicación

## 6.1.4 Tipo de Arquitectura

El campo de tipo de arquitectura tenía como objetivo identificar si la arquitectura propuesta es conceptual o concreta. El primero se refiere si los estudios presentan una arquitectura sin implementar la composición de servicios web semánticos, mientras que el último se refiere a composiciones de servicios web implementados en un prototipo siguiendo la arquitectura propuesta. A veces, estas arquitecturas concretas se presentaron juntas con resultados de comparación o con puntos de referencia de rendimiento. De la tabla 6.5 se puede ver que 28 estudios propusieron una arquitectura concreta. El resto de los estudios propuso una arquitectura conceptual. En las arquitecturas concretas algunas usan el framework java web para desarrollar aplicaciones semánticas (JENA) y otras el framework java para el desarrollo de agentes (JADE) o el entorno de desarrollo de agentes inteligentes JACK (JDE).

Tipo de arquitectura	Estudio	Conteo	Porcentaje
Concreta	[19],[21],[22],[23],[26],[27],[28],[29],[30],[32],[34],[35],[38],[39],[40],[43],[44],[46],[47],[52],[54],[55],[56],[59],[61],[66],[68],[69]	28	51.85
Conceptual	[18],[20],[24],[25],[31],[33],[36],[37],[41],[42],[45],[48],[49],[50],[51],[53],[57],[58],[60],[62],[63],[64],[65],[67],[70],[71]	26	48.15

Tabla 6.5 Estudios por tipo de arquitectura

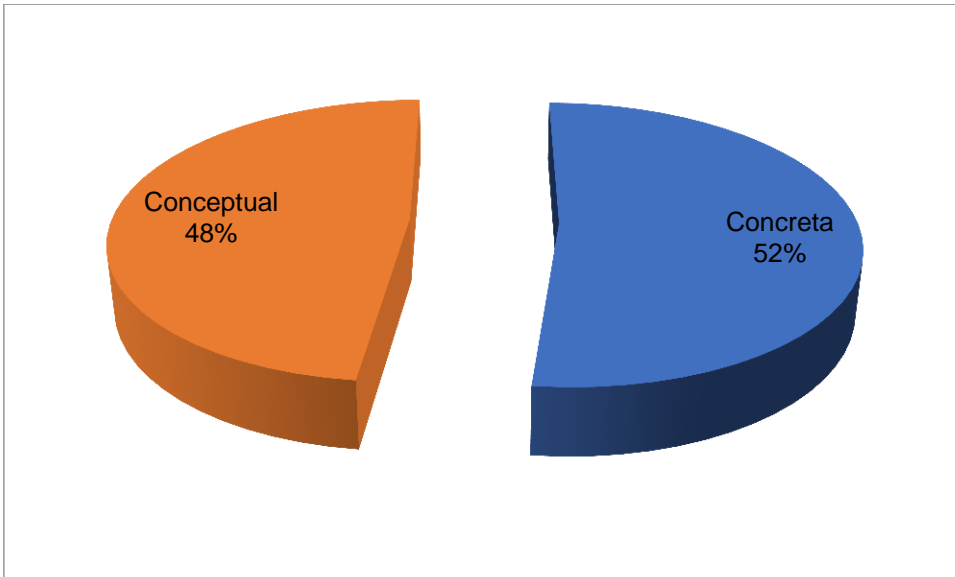


Figura 6.3 Tipo de Arquitectura

## 6.2 Respondiendo las preguntas de investigación

### 6.2.1 ¿Cuáles son los módulos de la arquitectura de los servicios web semánticos compuestos?

#### a) **Lista de los módulos que tiene la arquitectura.**

Los módulos de cada arquitectura mostrados en tabla 0.1 del apéndice, se pueden clasificar de acuerdo a su rol desempeñado, siendo los componentes identificados los siguientes:

1. Recolector. Este componente es el encargado de recolectar y filtrar servicios web que pertenezcan a un dominio específico.
2. Desplegador. Este componente es el encargado de desplegar los servicios en una red P2P. Una red P2P se encuentra organizado en varios grupos de pares. Los grupos contienen varios pares y servicios webs de cierto tipo. La misión del desplegador es ubicar un servicio web en su grupo correspondiente.
3. Interfaz de Usuario. Este componente se utiliza como un medio de ingreso de entradas de solicitudes y visualización de resultados. En el ingreso de solicitudes varía de acuerdo al nivel de automatización de la composición. En el nivel automático se utiliza para definir todo el flujo de la composición y los servicios web involucrados, en un nivel semiautomático se define anticipadamente solo el flujo de composición y en nivel automático el usuario solo ingresa sus requerimientos.
4. Analizador. Este componente abarca dos escenarios: la solicitud del servicio y la descripción del servicio. En ambos escenarios, su finalidad es obtener la información semántica de modo que pueda ser utilizada por el resto de componentes de la arquitectura. La información semántica puede estar obtenida de varios orígenes como por ejemplo desde un servicio web, una plantilla, un mensaje de texto o voz.
5. Buscador. Encargado de encontrar todos los servicios candidatos que puedan satisfacer la funcionalidad deseada expresada en la solicitud del servicio. Aquí se pueden dar dos situaciones. La primera situación es cuando la solicitud puede ser satisfecha por un solo

servicio en ese caso el servicio es inmediatamente enviada al módulo ejecutor, la segunda situación es cuando la solicitud solo puede ser satisfecha con una combinación de servicios, en este caso la solicitud es enviada a un compositor. En ambas situaciones este componente trabaja conjuntamente con un razonador semántico.

6. Mediador. Es el encargado de manejar problemas de interoperabilidad que surgen cuando se trabaja en un entorno heterogéneo. Los dos casos principales en los que se utiliza el mediador son: cuando los servicios web que intervienen en el proceso presentan diferentes formatos de mensaje, cuando los servicios web utilizan anotaciones de diferentes ontologías.
7. Razonador semántico .Es el encargado de encontrar la relación semántica entre los requerimientos del usuario y la descripción de los servicios web. En algunos estudios solo se emplean las entradas y salidas de los servicios web, y en muchos otros utilizan además de estas las precondiciones y efectos para garantizar una mayor exactitud en los resultados de correspondencia. El requerimiento del servicio puede darse también en forma de mensajes de texto o por la voz.
8. Traductor. Es el encargado de convertir las descripciones servicio web en un idioma entendible por el compositor, asimismo también se puede utilizar en un sentido inverso, para convertir las salidas generadas por un compositor en un lenguaje entendible por el resto de componentes de la arquitectura.
9. Compositor. Es el encargado de generar soluciones ante una solicitud de servicio que no pueda ser satisfecha por un servicio atómico. El compositor emplea un enfoque de descomposición del problema donde una tarea se puede subdividir en pequeñas tareas para poder cumplir con el objetivo deseado. El compositor utiliza diferentes técnicas de composición provenientes de la lógica lineal, lógica descriptiva, inteligencia artificial, enfoque dirigidos por modelos y modelos de representación transversales.
10. Validador externo. Es el encargado de validar si el flujo de los servicios en una combinación es razonable. Este componente asegura la exactitud del proceso de combinación dinámica utilizando lenguajes como Promela y herramientas como Spin.
11. Evaluador. Es el encargado de seleccionador la solución más óptima de entre varias soluciones generadas por el compositor, tomando en cuenta las propiedades no funcionales.
12. Ejecutor. Este componente presente tres características principales que son la invocación , personalización y extensibilidad :
  - Invocación, es el encargado de invocar los servicios web contenidos dentro de una solución final



- Personalización, permite generar descripciones de servicios compuestos en diferentes lenguajes de acuerdo a las preferencias del usuario , como por ejemplo: lenguaje de ejecución de procesos de negocio (BPEL), BPEL4WLS , XML , WSFL, XLANG
  - Extensibilidad, se puede agregar nuevos lenguajes de composición.
13. Adaptador. Es un componente encargado de seleccionar, reutilizar y adaptar una secuencia de servicios web atómicos agrupados como caso, de modo tal que satisfaga el requerimiento del usuario.
  14. Monitoreador. Es el encargado de supervisar y notificar el estado de los servicios y de la composición en tiempo de ejecución, además de recibir y ejecutar solicitudes de actualización en la base de conocimiento de calidad y la base de conocimiento en general. En el artículo [24] recibe el nombre de administrador de calidad, siendo este el encargado de almacenar y recuperar la información funcional y de calidad de servicio en registros UDDI y base de datos, siendo esta información proporcionada en el proceso de publicación de los proveedores, retroalimentación de los usuarios y supervisión de agentes inteligentes. En el estudio [36] este componente en coordinación con el reconfigurador permite manejar el indeterminismo en un entorno altamente dinámico.
  15. Reconfigurador. Es el encargado de abordar los cambios en la composición en tiempo de ejecución, cuando encuentra resultados no deseados, es decir cuando uno o varios servicios de la composición presentan fallas o deficiencias. Utiliza estrategias típicas de adaptación que incluyen la re-ejecución de servicios o re-planificación.
  16. Base de Conocimiento de Calidad. Es un contenedor donde se guarda base de datos y las ontologías de las propiedades no funcionales de los servicios web.
  17. Base de Conocimiento General. Es un contenedor donde se guarda principalmente las ontologías de dominio y las ontologías de servicio.
  18. Repositorio de Servicios. Es un contenedor donde se guardan los servicios web semánticos o sintácticos.
  19. Repositorio de reutilización. Permite almacenar composiciones y tablas de composiciones ya hechas exitosamente. Este componente permite mejorar los tiempos de ejecución y la escalabilidad reutilizando plantillas como por ejemplo modelos UML.
  20. Asignador de Tareas. Este componente define los servicios como tareas, y se encarga de asignar tareas a los robots en una manera más efectiva.
  21. Comunicador. Este componente es el encargado de apoyar la comunicación entre los robots, y la comunicación entre el robot y los componentes del lado del servidor.

22. Centro de Registro. Es un catálogo en la cual se puede publicar y encontrar referencias a los servicios web. Un ejemplo de centro de registro es un componente similar al UDDI.
23. Recuperador de Contexto. Es el encargado de recuperar la información del contexto desde la web y con esta información genera las condiciones que debe cumplir una solicitud de composición de servicios. Se entiende por contexto la información del entorno sobre el que un individuo interactúa.
24. Notificador. Este componente es el encargado de avisar al usuario que un servicio web se encuentra disponible y que puede acceder a este.
25. Diccionario. Este componente es utilizado en la interpretación de los requerimientos de los usuarios, para encontrar las intenciones, requerimientos funcionales importantes y restricciones. El diccionario está conformado por elementos léxicos relacionados con algunas entidades dentro de una ontología. Se utiliza en la técnica del procesamiento del lenguaje natural.
26. Coordinador. Es el encargado de coordinar la comunicación entre los componentes que forman parte de arquitecturas de diversa índole tales como arquitectura orientada a servicios y arquitectura de agentes abierta.

	Módulo	Estudio	Conteo	%
1	Recolector	[22]	1	0.24
2	Deployador	[22]	1	0.24
3	Interfaz de Usuario(GUI)	[23],[25],[27],[28],[29],[30],[39],[40],[41],[42],[47],[56],[57],[58],[61],[66],[67],[71]	18	4.24
4	analizador	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]	54	12.71
5	Buscador	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]	54	12.71
6	Mediador	[18],[20],[25],[42],[51],[61]	6	1.41
7	Razonador o Inferenciador	[18],[19],[20],[21],[22],[23],[24],[25],[28],[31],[32],[34],[35],[37],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]	46	10.82

8	Traductor	[18],[19],[23],[27],[29],[30],[33],[36],[39],[41],[42],[44],[45],[49],[50],[51],[53],[55],[59],[62],[64],[66],[67],[69],[71]	25	5.88
9	Compositor	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]	54	12.71
10	Validador Externo	[42],[43],[50],[69]	4	0.94
11	Evaluador	[21],[24],[39],[40],[43],[44],[49],[50],[52],[58],[60],[66],[67],[69]	14	3.29
12	Ejecutor	[18],[19],[20],[21],[22],[25],[27],[28],[33],[36],[38],[39],[41],[43],[44],[47],[48],[49],[50],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[63],[64],[65],[66],[68],[69],[70],[71]	37	8.71
13	Adaptador	[54]	1	0.24
14	Monitoreador	[24],[32],[33],[36],[38],[39],[47],[50],[52],[58],[60],[65]	12	2.82
15	Reconfigurador	[32],[33],[36],[39],[52],[58],[60],[65]	8	1.88
16	Base de Conocimiento de Calidad	[21],[24],[43],[50],[58]	4	0.94
17	Base de conocimiento General	[25],[26],[27],[28],[29],[30],[33],[34],[35],[36],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]	44	10.35
18	Repositorio de Servicios	[21],[24],[26],[29],[30],[31],[39],[48],[49],[57],[60],[62],[63],[69],[71]	15	3.53
19	Repositorio de reutilización	[20],[21],[47],[48],[49],[50],[53],[54],[55],[57],[58],[59],[68]	13	3.06
20	Asignador de Tareas	[19]	1	0.24
21	Comunicador	[19]	1	0.24
22	Centro de Registro	[48],[50],[60],[65],[69],[71]	6	1.41
23	Recuperador de Contexto	[55]	1	0.24
24	Notificador	[63]	1	0.24
25	Diccionario	[64]	1	0.24
26	Coordinador	[24],[25],[67]	3	0.71

**Tabla 6.6 Lista de módulos**

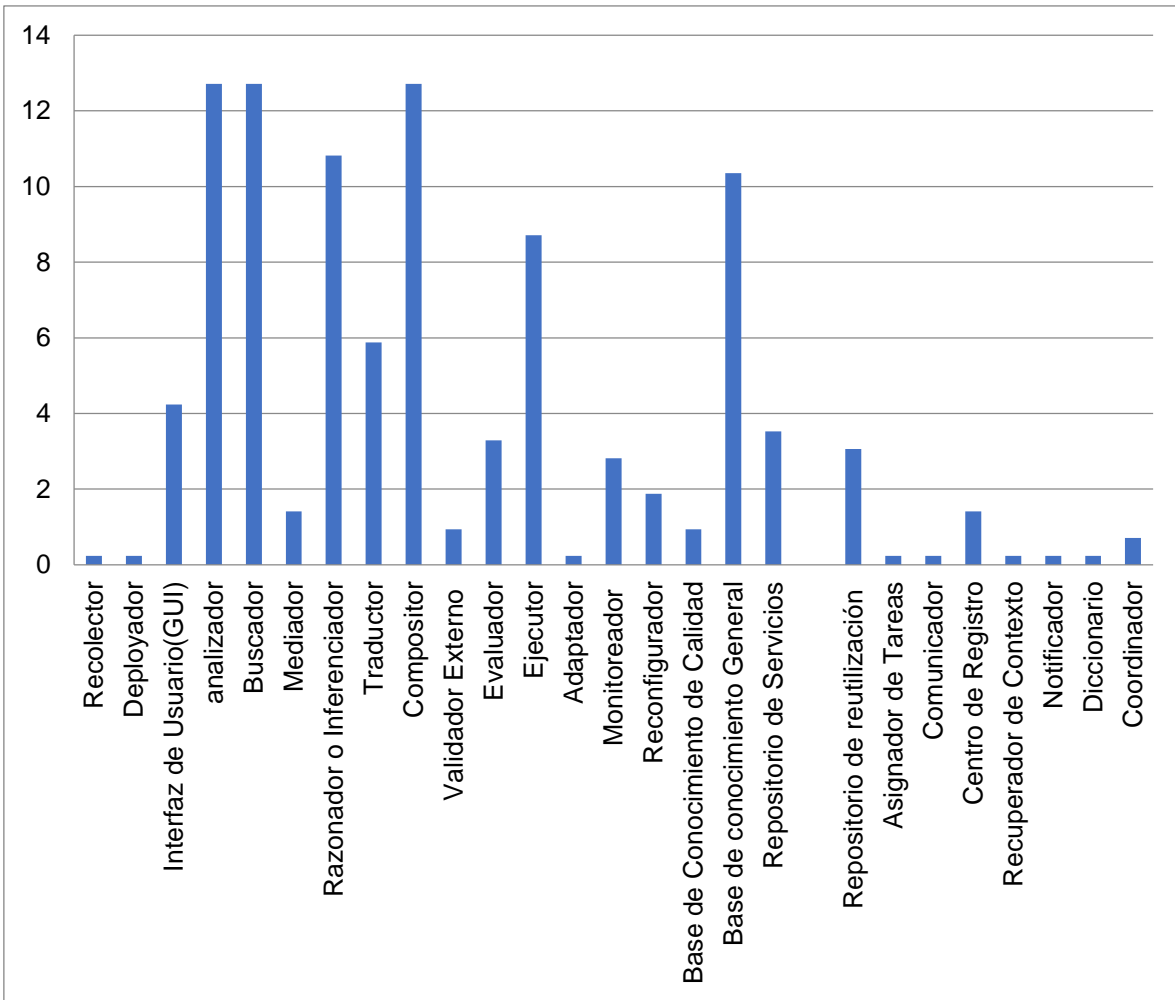


Figura 6.4 Componentes de arquitectura

a) **Lista de patrones de arquitectura**

En este punto es necesario distinguir lo que significa un patrón de arquitectura con respecto a un estilo de arquitectura. El patrón se enfoca en resolver un problema recurrente dentro de un contexto específico, mientras que el estilo de arquitectura se centra en los componentes, sus interacciones y las restricciones, con unas breves indicaciones sobre cuando es conveniente utilizarlo sin enfocarse en resolver un problema. En la revisión sistemática todos los estudios presentan una arquitectura de N-Capas, en la tabla 6.7 se muestra los resultados obtenidos.

Patrón	Estudio	Conteo	%
N-Capas	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30] [31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43] [44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56] [57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69] [70],[71]	54	100

Tabla 6.7 Patrón de Arquitectura.

En cuanto estilos arquitectónicos , se puede identificar que la mayoría de los estudios mostrados en la tabla 6.8 , utiliza el estilo de arquitectura orientada a servicios (SOA) , donde los componentes tienden a ser servicios, los cuales contienen la funcionalidad de negocio y se integran en forma flexible debido a que estos son débilmente acoplados e independientes de la plataforma . Otros estudios tienden a combinar dos estilos arquitectónicos tales como la arquitectura orientada a servicios (SOA) y las redes entre pares (P2P). En las redes entre pares se encontró el uso de un algoritmo epidémico, este protocolo de red permite la rápida difusión de un mensaje desde una fuente hacia los nodos vecinos ampliándose por toda la red. La tabla 6.8 se muestra un cuadro resumido de los estilos arquitectónicos encontrados:

Estilo	Estudios	Conteo	%
SOA	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34], [35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51], [52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68], [69],[70],[71]	54	93.10
P2P	[21],[22],[47],[65]	4	6.90

Tabla 6.8 Estilos de arquitectura.

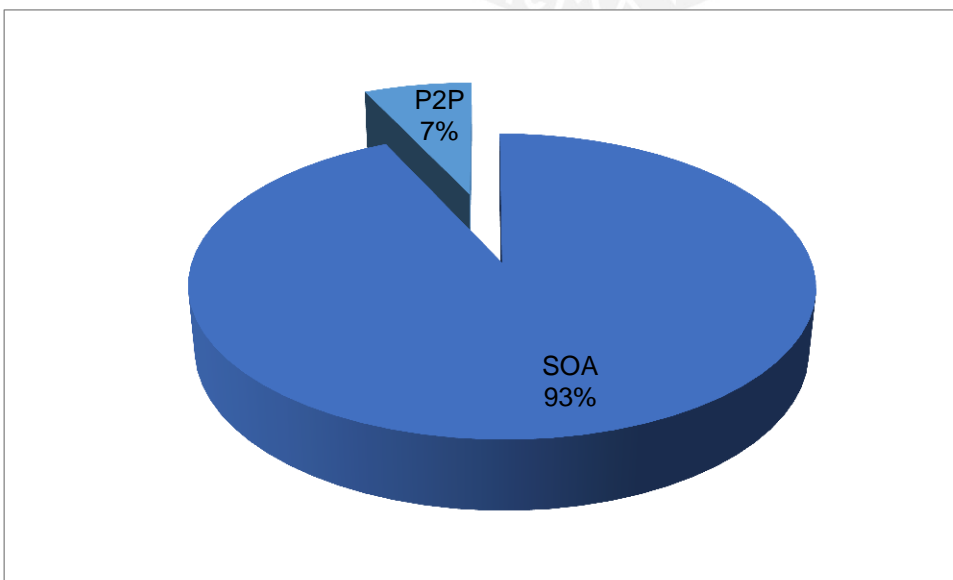


Figura 6.5 Patrón de arquitectura

## b) Nivel de automatización

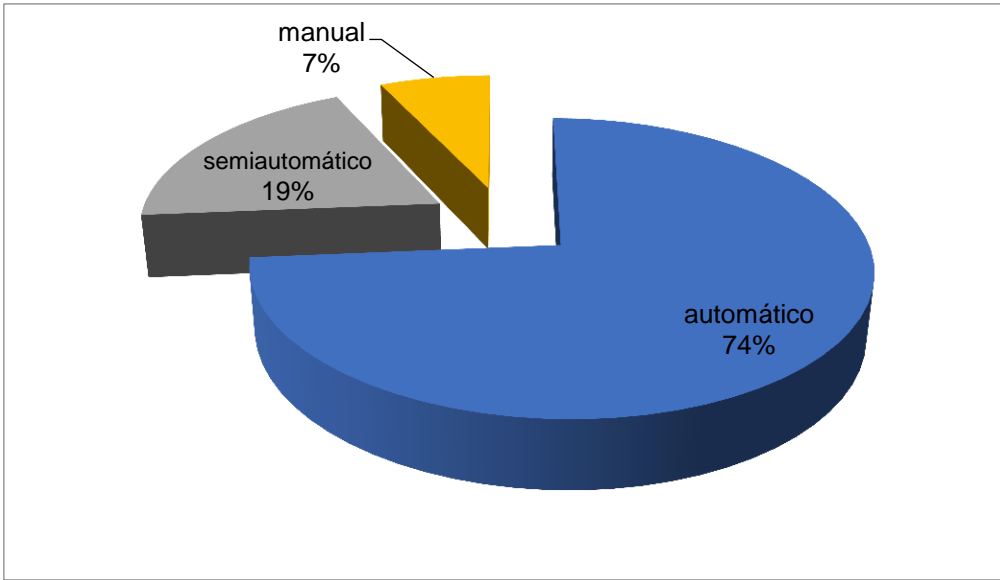
En el nivel de automatización se muestra que la mayoría tienden a una composición automática, esto se explica debido a que este nivel es requerido en un entorno real. A continuación se explica los niveles de automatización:

- a) Manual. Cuando el flujo de ejecución y la referencia a los servicios web semánticos es efectuado por medio de la intervención de una persona. Esta propuesta tiende a ser proclive a errores, debido a la gran cantidad de servicios web existentes y por publicar por diferentes proveedores de servicios, lo que da como resultado una pérdida de flexibilidad y falta de aprovechamiento de los servicios web existentes. [25],[39],[48],[61]
  
- b) Automático. En este enfoque las actividades manuales son reemplazados por un algoritmo que permiten la composición automatizada a través de un plan de composición [18],[19],[20],[21],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[43],[44],[45],[46],[47],[49],[51],[52],[55],[58],[59],[60],[62],[63],[64],[66],[67],[68],[69],[70]
  
- c) Semiautomático. En este enfoque se puede dar dos casos. La primera donde el descubrimiento y selección de los servicios web tienden a seguir un proceso automatizado para poder efectuar una composición manual, la segunda donde el descubrimiento de los servicios tienden a seguir un proceso manual siendo la composición automatizada pero restringida a los servicios seleccionados [22],[25],[39],[41],[42],[50],[53],[54],[56],[57],[71]

En la tabla 6.9 se muestra un cuadro resumido de los niveles de automatización encontrados en los artículos de investigación. El artículo [39] presenta una arquitectura adaptable que soporta diferentes tipos de niveles de automatización, es por eso que se le referencia en los 3 niveles.

Nivel de Automatización	Estudios	Conteo	%
automático	[18],[19],[20],[21],[23],[24],[25],[26],[27],[28],[29],[30],[31] [32],[33],[34],[35],[36],[37],[38],[39],[40],[43],[44],[45],[46], [47],[49],[51],[52],[55],[58],[59],[60],[62],[63],[64],[66],[67], [68],[69],[70]	42	73.68
semiautomático	[22],[25],[39],[41],[42],[50],[53],[54],[56],[57],[71]	11	19.30
manual	[25],[39],[48],[61]	4	7.02

Tabla 6.9 Nivel de automatización

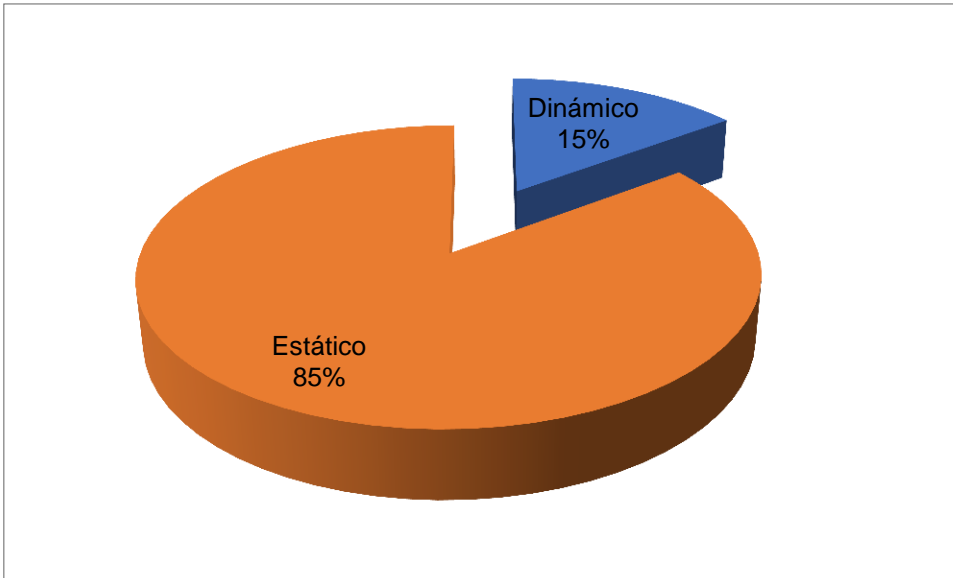


**Figura 6.6 Nivel de automatización**

Además de estos enfoques se puede categorizar la composición como dinámicas o estáticas, las composiciones dinámicas tienen la característica de replantear un plan de composición cuando un servicio web fue dado de baja por un proveedor o cuando ocurrió un error que se presentó cuando la composición ya se encontraba ejecutando .Las composiciones estáticas es cuando el sistema no tiene la capacidad de una reconfiguración dinámica en tiempo de ejecución o necesita de la intervención humana para subsanar el problema .

Nivel de Respuesta	Estudios	Conteo	%
Dinámico	[32],[33],[36],[39],[52],[58],[60],[65]	8	14.81
Estático	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[34],[35],[37],[38],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[53],[54],[55],[56],[57],[59],[61],[62],[63],[64],[66],[67],[68],[69],[70],[71]	46	85.19

**Tabla 6.10 Nivel de dinamismo**

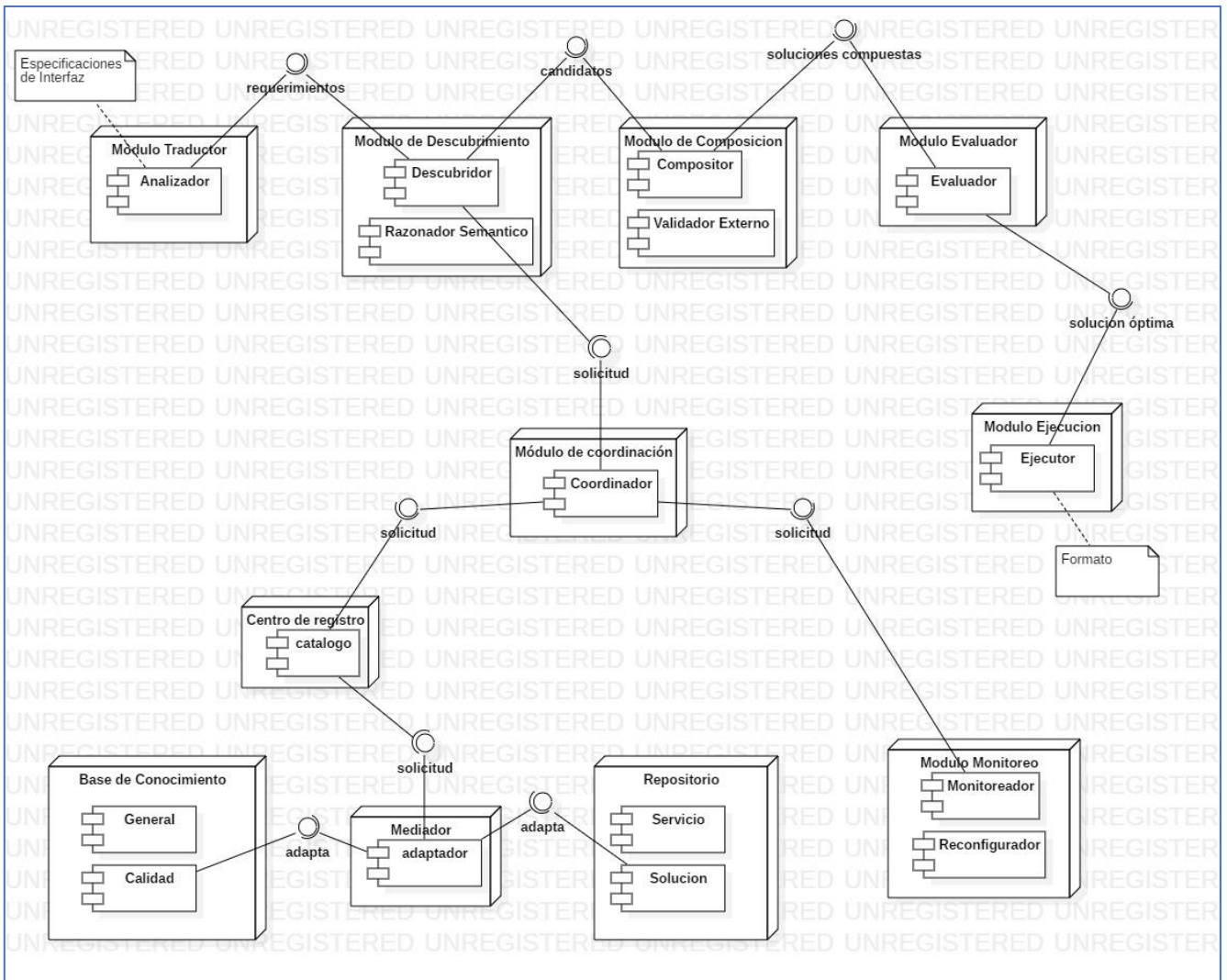


**Figura 6.7 Nivel de Dinamismo**

#### **6.2.1.1 Análisis y discusión**

Tomado los resultados de la tabla 6.6 y del gráfico 6.4, se presenta una arquitectura genérica, tomando los componentes más importantes de acuerdo a su valor mostrado en el campo conteo o porcentaje. Los componentes cuyos valores de incidencia fueron mayores a 1 son los siguientes: analizador, buscador, compositor, razonador, base de conocimiento general, ejecutor, traductor, interfaz de usuario, repositorio de Servicios, evaluador, repositorio de reutilización, monitoreador, reconfigurador, mediador, centro de registro, validador externo, base de conocimiento de calidad, coordinador. Estos módulos son mostrados en la figura 6.8.





**Figura 6.8** Arquitectura de servicios web compuestos semánticos

Si se divide los componentes seleccionados en dos grandes grupos se obtiene lo siguiente

- Grupo A: se encuentra conformado por el analizador, buscador, compositor, razonador, base de conocimiento general, ejecutor, traductor, interfaz de usuario
- Grupo B: se encuentra conformado por el evaluador, repositorio de reutilización, monitoreador, reconfigurador, mediador, centro de registro, validador externo, base de conocimiento de calidad, coordinador.

Por lo tanto, se puede deducir que existen desafíos en la composición que no son muy abordados en los artículos seleccionados pero que son muy importantes en el desarrollo una arquitectura robusta, los desafíos de la arquitectura son las siguientes:

- Capacidad para trabajar en un entorno heterogéneo.
- Capacidad para trabajar en un entorno altamente volátil o indeterminado.
- Capacidad en encontrar soluciones óptimas tomando en cuenta requerimientos no funcionales y técnicas de validación de la composición de servicios web.
- Incrementar el rendimiento en la generación de la soluciones reutilizando soluciones exitosas
- Visión holística de la arquitectura con una gran fluidez en la comunicación de los componentes que lo conforman.



## 6.2.2 ¿Cómo se utilizan las ontologías en una arquitectura de servicios web semánticos compuestos?

### a) Lista de Lenguaje de Ontologías por Estudio

Según la información mostrada en la tabla 0.2 del apéndice , la mayoría de las propuestas tienden a usar lenguajes estandarizados de ontología tales como DAML-S y DAML [23],[57],[66],[67],[69], OWL y OWL-S [19],[21],[22],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[40],[43],[44],[45] , [46],[47],[48],[49],[50],[51],[52],[53],[55],[56],[59],[60],[62],[63],[64],[65],[68],[69],[70],[71] y OWL-DL [53], es necesario anotar que los lenguajes DAML, DAML-S han sido reemplazados por OWL y OWL-S respectivamente. En menor escala existen estudios que utilizan WSMO [18],[37],[39],[41],[43],[53], [61],[65],[70], WSPO[41] que se encuentran dentro de la familia WSML . En este grupo también se utiliza el lenguaje SAWSDL que permite hacer anotaciones semánticas a documentos WSDL [48],[59],[61]. Algunos propuestas efectúan una extensión de estas, tal es el caso del lenguaje OWL-SC que es una ontología OWL-S al que se agregado el contexto [45],[55] y de la ontología CBROnto que está basado en OWL [54]. También se pueden mencionar otras ontologías que se puede calificar como propias tales como ISCF Task [25], iRPG[48] permite hacer anotaciones semánticas a los servicios existentes. Por último se encontraron estudios en donde solo menciona el uso de una ontología pero sin dar detalle sobre el nombre [20][24][42][58] Los estudios que no utilizan lenguajes de ontologías .

Ontología	Estudios	Conteo	%
DAML-S y DAML	[23],[57],[66],[67],[69]	5	7.35
OWL y OWL-S	[19],[21],[22],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[40],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[55],[56],[59],[60],[62],[63],[64],[65],[68],[69],[70],[71]	40	58.82
OWL-DL	[53]	1	1.47
OWL-SC	[45][55]	2	2.94
WSMO	[18],[37],[39],[41],[43],[53],[61],[65],[70]	9	13.24
WSPO	[41]	1	1.47
SAWSDL	[48],[59],[61]	3	4.41
CBROnto	[54]	1	1.47
ISCF Task	[25]	1	1.47

iRPG	[48]	1	1.47
Sin definir	[20],[24],[42],[58]	4	5.88

Tabla 6.11 Lenguaje de Ontología

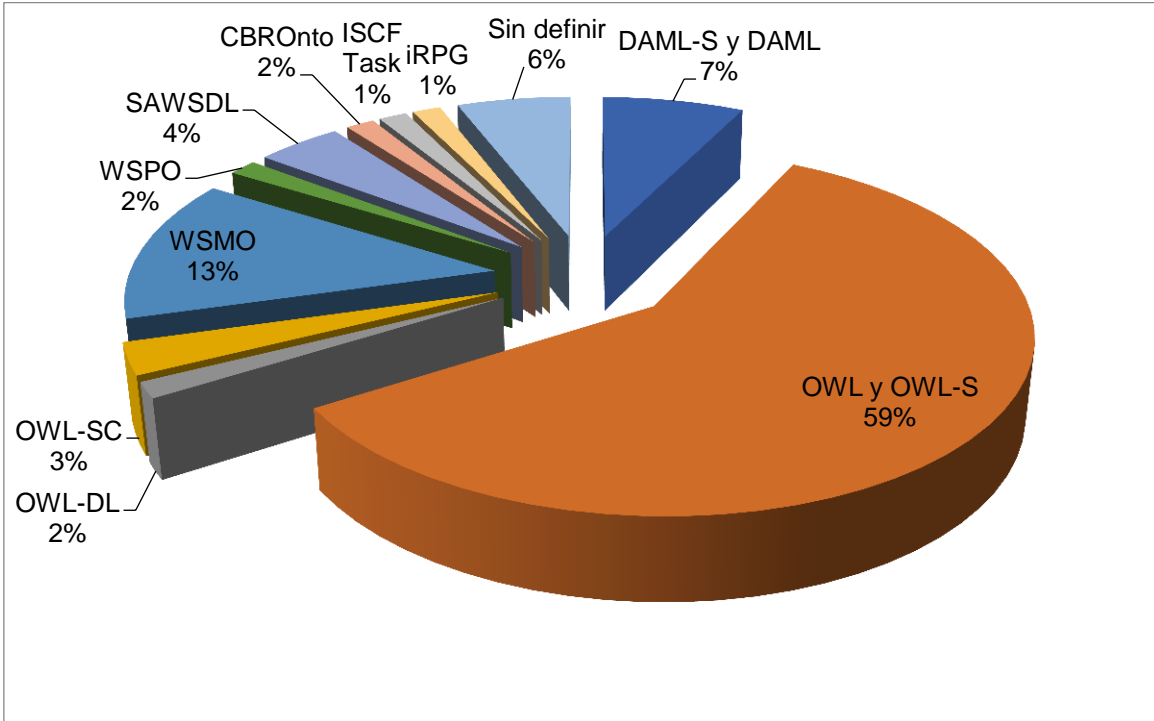


Figura 6.9 Lenguaje de ontología

## b) Etapa en la que interviene

Las etapas en donde se usa la ontología son las siguientes:

1. Fase de Especificación.- Se da cuando se define una ontología que pueda ser compartida y entendida por los sistemas involucrados [18],[19],[20],[21],[22],[23],[24],[25],[26], [27],[28], [29], [30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50], [51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]
2. Fase de Búsqueda .-Se da cuando se busca servicios atómicos que permitan satisfacer la solicitud de servicio [18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29], [30],[31],[32],[33],[34], [35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55], [56], [57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]
3. Fase de Composición.-Se da cuando un servicio atómico existente no puede satisfacer la solicitud de servicio , por tal motivo se genera composiciones de servicios web , para satisfacer la solicitud de servicio[18],[19],[20],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[34],[35]

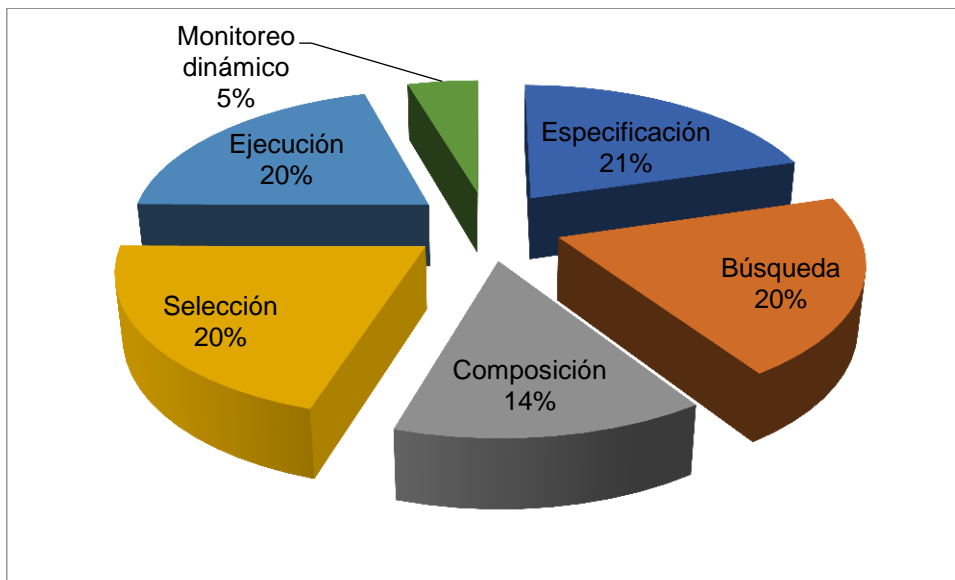
[37],[40],[41],[42],[44],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[60],[62],[63],[64],[65],[68],[69],[70]

4. Fase de Selección.-Se da cuando se presentan una gran cantidad de servicios web que satisfacen los requerimientos funcionales de la solicitud, por tal motivo se necesita discriminar y seleccionar los mejores que puedan satisfacer las preferencias, restricciones de solicitante. [18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]
5. Fase de Ejecución.-Se da cuando se dispone de la lista final de los servicios seleccionados y se comienza a referenciarlos para su ejecución [18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[63],[64],[65],[66],[67],[68],[69],[70],[71]
6. Fase Monitoreo dinámico. Es cuando se actualiza los modelos ontológicos en respuesta a un evento [24],[32],[33],[36],[38],[39],[47],[50],[52],[58],[60],[65]

Etapa	Estudios	Conteo	%
Especificación	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32] [33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47] [48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62] [63],[64],[65],[66],[67],[68],[69],[70],[71]	54	20.38
Búsqueda	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32] [33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47] [48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62] [63],[64],[65],[66],[67],[68],[69],[70],[71]	54	20.38
Composición	[18],[20],[21],[22],[25],[28],[29],[30],[31],[32],[34],[35],[37],[38],[40] ,[41],[44],[46],[47],[48],[49],[50],[51],[53],[54],[56],[57],[59],[60],[61] ,[63],[64],[65],[68],[69],[70],[71]	37	13.96
Selección	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32] [33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47] [48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62] [63],[64],[65],[66],[67],[68],[69],[70],[71]	54	20.38

Ejecución	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32] [33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47] [48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62] [63],[64],[65],[66],[67],[68],[69],[70],[71]	54	20.38
Monitoreo dinámico	[24],[32],[33],[36],[38],[39],[47],[50],[52],[58],[60],[65]	12	4.53

**Tabla 6.12 Etapa en la que interviene la ontología**



**Figura 6.10 Etapa en la que interviene la ontología**

### c) Propósito de la Ontología

Se tienen los siguientes tipos de ontologías, los cuales permiten establecer en forma clara el propósito de la Ontología

1. Ontología de Dominio.- La ontología de dominio, nos permite tener una base de conocimiento sobre un dominio especializado en el que va funcionar un sistema, siendo este conocimiento basado en conceptos, taxonomías, relaciones y reglas. Por ejemplo en los estudios seleccionados se encuentra estudios que funcionan en el ámbito de sistemas de fabricación, salud y gobierno entre otros, siendo esos campos denominados dominios.

[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[39],[40],[42],[43],[44],[45],[46],[47],[48],[49],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[64],[65],[66],[67],[68],[69],[71]

2. Ontología de Servicio.- Esta ontología tiene la finalidad de describir y compartir la información funcional de un servicio web utilizando conceptos y sus relaciones en las áreas específicas del servicio web. Esta ontología provee información importante para saber lo que proporciona el servicio, su uso y como interactuar con él. [18],[19],[20],[21],[22],[23],[24],[25],[27],[28],[29],[30],[31],[32],[33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62],[64],[65],[66],[67],[68],[69],[70],[71]
3. Ontología de tarea.- Esta ontología describe el dominio de un problema de planeamiento incluye una colección de operadores y métodos de planeamiento. Se denomina también como una ontología de Lógica descriptiva. [18]
4. Ontología de Calidad.- Especifica vocabularios y conocimiento compartido sobre la calidad de las propiedades de servicio y su relación con los servicios web semánticos [43],[58]
5. Ontología de Perfil de Usuario.-Esta ontología tiene la finalidad de describir, compartir y representar las solicitudes y/o preferencias del usuario como conceptos.[46],[57],[58]
6. Ontología de Meta .- Esta ontología cubre aspectos relacionados con respecto a la funcionalidad solicitada [18],[26]
7. Ontología de Transacción. Esta ontología permite definir las transacciones de un servicio web , una transacción puede ser de 5 tipos : entrega , descubrimiento ,invocación, composición y flujo de trabajo[25]
8. Ontología de Contexto.- Toman en cuenta la información de contexto para hacer las adaptaciones necesarias.[45],[55]
9. Ontología del Evento.- Describe un evento que es provocado por una condición de dominio[46]
10. Ontología del Agente BDI.-Describe conceptos tales como : la meta que el agente quiere alcanzar ,las creencias , y el plan que el agente ejecutará [46]
11. Ontología de dominio sobre patrones de flujo de trabajo. Define un flujo de trabajo sobre una tarea en un dominio.[46]
12. Ontología común.- Define el conocimiento de la vida diaria humana , se utiliza para traducir y comprender el requerimiento del solicitante del servicio[49]
13. Ontología CBR. Es una ontología para describir la estructura de los casos de un sistema CBR(Razonamiento basado en casos) [54]
14. Ontología BR .Es una ontología que describe las reglas de negocio [70]

Propósito	Estudios	Conteo	%
Ontología de Dominio	[18],[19],[20],[21],[22],[23],[24],[25],[26],[27],[28],[29],[30],[31] [32],[33],[34],[35],[36],[37],[39],[40],[42],[43],[44],[45],[46],[47] [48],[49],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61],[62] [64],[65],[66],[67],[68],[69],[71]	49	41.88
Ontología de Servicio	[18],[19],[20],[21],[22],[23],[24],[25],[27],[28],[29],[30],[31],[32] [33],[34],[35],[36],[37],[38],[39],[40],[41],[42],[43],[44],[45],[46] [47],[48],[49],[51],[52],[53],[54],[55],[56],[57],[58],[59],[60],[61] [62],[64],[65],[66],[67],[68],[69],[70],[71]	51	43.59
Ontología de tarea.	[18]	1	0.85
Ontología de Calidad	[43],[58]	2	1.71
Ontología de Perfil de Usuario	[46],[57],[58]	3	2.56
Ontología de Meta	[18],[26]	2	1.71
Ontología de Transacción	[25]	1	0.85
Ontología de Contexto	[45],[55]	2	1.71
Ontología del Evento	[46]	1	0.85
Ontología del Agente BDI	[46]	1	0.85
Ontología de Flujo de Trabajo	[46]	1	0.85
Ontología común	[49]	1	0.85
Ontología CBR	[54]	1	0.85
Ontología BR	[70]	1	0.85

Tabla 6.13 Propósito de la ontología



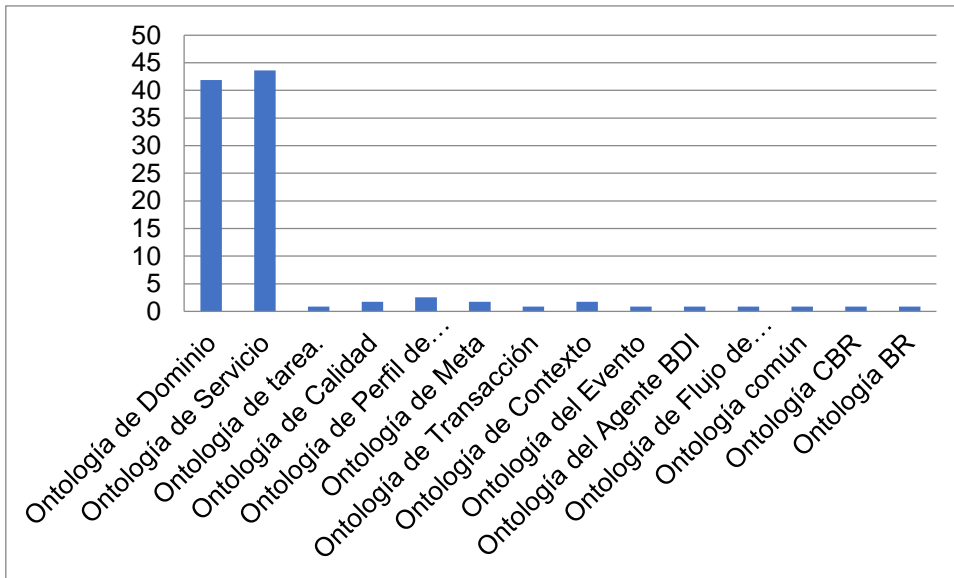


Figura 6.11 Propósito de la ontología

### 6.2.2.1 Análisis y Discusión

El uso de ontologías es fundamental en la arquitectura, las ontologías se emplean para los siguientes propósitos:

- Permite describir los servicios web y del entorno en un formato que se pueda compartir. Las ontologías más utilizadas son las ontologías de dominio y de contexto.
- Permite interpretar las solicitudes de servicio en un formato entendible para un entorno automatizado.
- Permitir el descubrimiento, composición, evaluación y ejecución automatizada debido a que en la actualidad se utilizan servicios web sintácticos que no pueden ser entendidas por un agente además las búsquedas sintácticas no brindan exactitud ya que solo permiten una búsqueda basada en palabras claves que induce a falsos positivos, es decir se encuentra servicios web que satisfacen la solicitud del usuario sin embargo son rechazados. De acuerdo a la información presentada en la figura 6.10 no todas los estudios utilizan el razonamiento semántico para la composición de servicios en su lugar usan otras técnicas tales como el planeamiento de la inteligencia artificial.
- Permitir el razonamiento sobre las propiedades del servicio. De acuerdo a lo analizado en los estudios se encontró dos tipos de relación:
  - Relación basada en categoría. Se evalúa la correspondencia entre la solicitud y los servicios web existentes arriba. Los niveles de correspondencia se clasifica en cuatro categorías

(exacto, contenido, contenedor, falla) o en cinco categorías (exacto, contenido, contenedor, intersección, nulo). Los conceptos que presentan un mayor nivel de correspondencia son los que presentan el grado de coincidencia exacto y contenido en ese orden.

- Relación basada en Clasificación. Se utiliza un cálculo de similitud entre los conceptos arriba descritos para cuantificar la relación entre los conceptos y con los valores de similitud obtenida, se establece una clasificación.

Los estudios [35],[39] efectúan una composición funcional , en donde los niveles de correspondencia no solo abordan conceptos de entrada y salida de los servicios web si no también agregan las pre y postcondiciones.



## 6.2.3 ¿Cómo soportar las capacidades de seguridad en una arquitectura de servicios web semánticos compuestos?

### a) Etapa en la que interviene los requerimientos de seguridad.

En la tabla 0.1 del apéndice, se lista todos los componentes de arquitectura de todos los estudios seleccionados obtenidos durante la ejecución de protocolo de revisión. Para identificar la utilización de los requerimientos de seguridad en cada arquitectura, se analizó cada componente y la fase en la cual este componente intervenía. De acuerdo a la información obtenida se identificó las siguientes etapas:

1. Fase de especificación. En los estudios [21],[24],[43],[55],[58],[66],[67] se utilizan los requerimientos de seguridad para realizar las descripciones semánticas de los servicios.
2. Fase de evaluación. En los estudios [21],[24],[39],[40],[43],[44],[49],[50],[52],[58],[60],[66],[67],[69] se utilizan los requerimientos de seguridad para seleccionar el plan más óptimo. En esta fase se resalta los siguientes estudios: El estudio [39] utiliza técnicas sofisticadas de QoS para la evaluación y selección, el estudio [58] efectúa un ranking de los planes generados, siendo el plan con el valor más alto cercano a las preferencias del usuario el que se enviará a la fase de ejecución, el estudio [69], utiliza tres parámetros de calidad de composición: ranking de la composición, relevancia de la composición y completitud de la composición
3. Fase Monitoreo dinámico. Es cuando existe una actualización de las propiedades no funcionales en especial las de tipo dinámica como resultado de una retroalimentación con el usuario o por medio de un agente. En los estudios [24],[50],[58],[60] ocurren acciones de monitoreo.

Etapa	Estudios	Conteo	%
Especificación	[21],[24],[43],[55],[58],[66],[67]	7	28
Evaluación	[21],[24],[39],[40],[43],[44],[49],[50],[52],[58],[60],[66],[67],[69]	14	56
Monitoreo dinámico	[24],[50],[58],[60]	4	16

Tabla 6.14 Etapa en la que interviene los requerimientos de seguridad

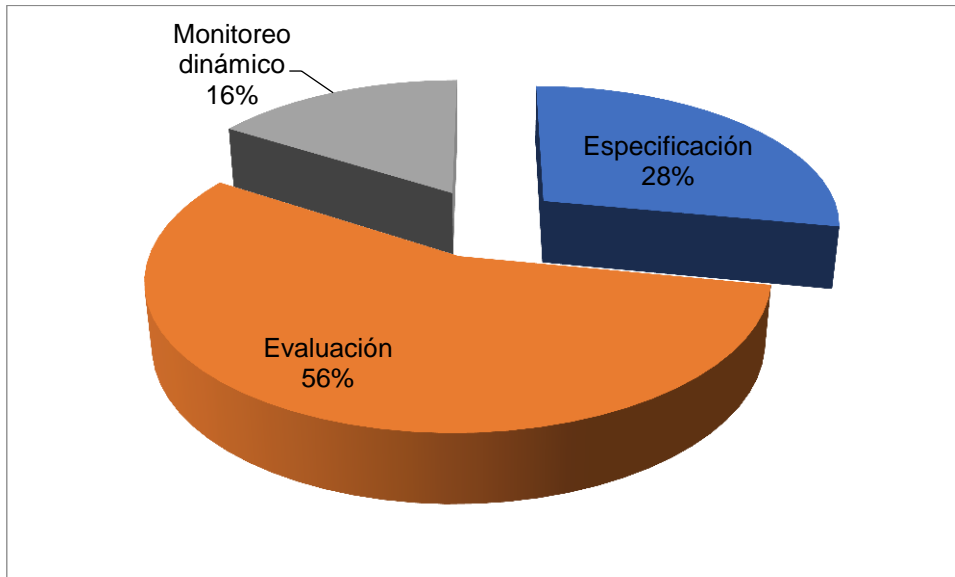


Figura 6.12 Etapa en la que interviene

## b) Componente de la seguridad.

1. Base de Datos QoS. En los estudios [21], [24], [50] se presenta una base de datos QoS que almacena la información de calidad del servicio. El estudio [24] utiliza dos tipos de tablas , una tabla para almacenar la información publicada por los proveedores de los servicios web , y otra tabla para almacenar la información proporcionada por un agente de monitoreo y la retroalimentación con el usuario, el artículo hace esta separación porque clasifica los atributos de calidad de dos tipos: atributos estáticos y atributos dinámicos , los atributos estáticos son aquellos en los cuales sus valores se mantienen como por ejemplo el costo ,

los atributos dinámicos como el tiempo de respuesta y confianza tienden a variar dependiendo de la retroalimentación con el usuario y del agente de monitoreo.

2. **Administrador de Calidad.** En el estudio [24] se presenta este componente , que tiene dos funciones principales :
  - a. Almacenar la información de la calidad de servicio (QoS) de los servicios web semánticos.
  - b. Actualizar de la información de calidad, la cual es recolectada a través de la retroalimentación con los usuarios y por medio de agentes de monitoreo.
  
3. **Componente Descubridor.** En el estudio [50] se consulta a una base de datos para buscar los servicios.
  
4. **Componente Ejecutor.** En el estudio [50] se efectúa una retroalimentación a la base de datos de calidad de servicio (QoS)
  
5. **Componente Evaluador.** En los estudios [24],[39],[40],[43],[44],[49],[52],[58],[60],[69] se presenta un componente evaluador que toma en cuenta las propiedades no funcionales de los servicios web que componen el plan de solución y tiene como salida el mejor plan de solución. En el estudio [24] se le denomina administrador de ontología. En el estudio [39] solo menciona un módulo evaluador. En el estudio [40] se le denomina modulo optimizador. En el estudio [44] busca una solución óptima utilizando expresiones de restricción que se originan a partir de las propiedades no funcionales. En el estudio [49] se le denomina componente recomendador y evaluador que permite seleccionar el mejor modelo de composición entre varios modelos. En el estudio [60], El agente de planeamiento utiliza una técnica de selección basado en propiedades no funcionales. En el estudio [69] Se selecciona un plan basado en parámetros de calidad de composición , se define tres parámetros de calidad de composición : ranking, relevancia y exhaustividad
  
6. **Ontología.** En el estudio [55] se define una ontología OWL-SC que toma en cuenta el contexto. La categoría W-Contexto abarca un número de propiedades no funcionales de los servicios. En el estudio [58] se presenta una ontología de QoS donde se almacena un vocabulario y conocimiento compartido de las propiedades de la calidad de servicio de los servicios web. El estudio [24], [43] utiliza una ontología de dominio que toma en cuenta propiedades funcionales y no funcionales. En el estudio [66],[67] los atributos no funcionales se encuentran en el perfil del servicio del DAML-S

7. Componente traductor. En el estudio [66] el traductor y el probador del teorema de lógica lineal, clasifican los atributos no funcionales en tres categorías: atributos cuantitativos consumibles, restricciones cualitativas y resultados cualitativos. En el estudio [67], el parseador DAML-S y el planeador de lógica lineal, clasifican los atributos no funcionales en cuatro categorías: atributos cuantitativos consumibles, atributos cuantitativos no consumibles, restricciones cualitativas y hechos cualitativos. En el estudio [43], se presenta el motor de transformación que identifica las propiedades no funcionales de la solicitud del usuario y los transforma en expresiones de restricción, en este estudio se clasifica los atributos QoS en dos categorías:
- a. Estáticas. Estos atributos no se modifican en tiempo de ejecución y son definidos por los proveedores de servicios, en esta categoría se puede encontrar los siguientes atributos: escalabilidad, seguridad.
  - b. Dinámicas. Estos atributos se modifican en tiempo de ejecución y son definidos en forma abstracta en la solicitud de los usuarios, en esta categoría se puede encontrar los siguientes atributos: disponibilidad, tiempo de respuesta, rendimiento.

Componente	Estudios	Conteo	%
Base de Datos QoS	[21], [24], [50]	3	12
Administrador de Calidad.	[24]	1	4
Componente Descubridor	[50]	1	4
Componente Ejecutor	[50]	1	4
Componente Evaluador	[24],[39],[40],[43],[44],[49],[52],[58],[60],[69]	10	40
Ontología	[24],[43],[55],[58],[66],[67]	6	24
Componente traductor	[43],[66],[67]	3	12

**Tabla 6.15 Componente de Seguridad**

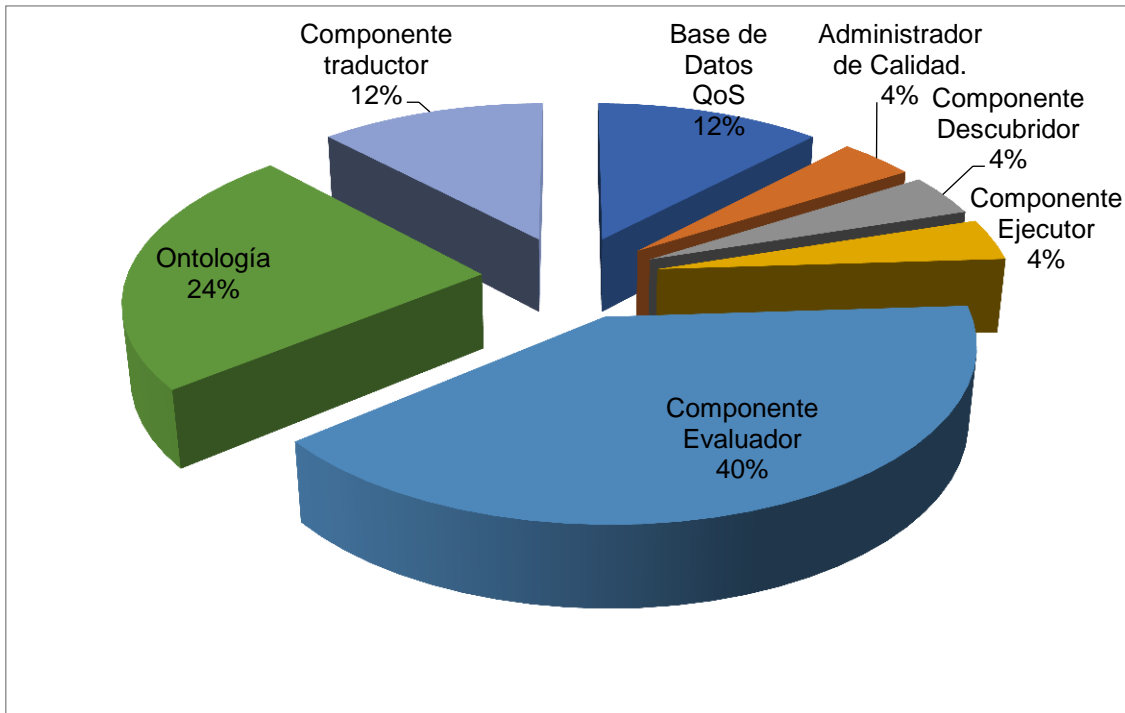


Figura 6.13 Componente de seguridad

### 6.2.3.1 Análisis y Discusión

En los estudios seleccionados se encontró tres tipos de evaluación usados en la composición de servicios:

- La selección por el grado de coincidencia semántica.
- La selección por el cálculo de similitud, hay algunos artículos que solo buscan una coincidencia en las entradas y salidas de la solicitud del usuario, otras buscan además de las mencionadas, el cumplimiento de las precondiciones y efectos.
- La selección por el cumplimiento de los requerimientos no funcionales. Estas se basan en el grado de cumplimiento con las propiedades calidad.

La seguridad es tratada como requerimiento no funcional y se utiliza para reducir los espacios de búsqueda y para poder encontrar la solución más óptima de entre varias alternativas, en algunos artículos se trata a la seguridad como una propiedad dinámica y en otras como estática, sin embargo es necesario que estén almacenados en forma persistente en base de datos y en ontologías de modo tal que sean rápidamente accedidos y actualizados por otros componentes de la arquitectura en forma permanente.

## 6.2.4 ¿Cuáles son las técnicas de composición en una arquitectura de servicios web semánticos compuestos?

### a) Campo de aplicación al que pertenece el criterio utilizado

El campo de aplicación al que pertenecen los criterios utilizados en la composición de acuerdo a los resultados mostrados en la tabla 0.3 del apéndice se pueden clasificar en nueve categorías:

1. Razonamiento sobre Ontologías. Se refiere a que se efectúa el razonamiento utilizando el cálculo de la similitud entre los conceptos en una ontología. Algunos estudios utilizan las propiedades de entradas y salidas de los servicios web semánticos, otros estudios efectúan una composición funcional utilizando las precondiciones y efectos además de los parámetros mencionados anteriormente [21],[22],[23],[24],[25],[28],[31],[32],[34],[35],[37],[38],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[63],[64],[65],[66],[67],[68],[69],[70],[71] muestran técnicas contenidas en esta categoría.
2. Razonamiento Lógico. Se refiere al uso de conceptos de la lógica lineal para resolver un problema de composición [19],[23],[24],[26],[62],[66],[67]
3. Razonamiento Híbrido. Se refiere a la combinación de técnicas de diferentes campos de investigación para lograr la composición [18]
4. Razonamiento Basado en Casos. Es una técnica donde se resuelven nuevos problemas reutilizando adaptando soluciones exitosas a problemas similares [54]
5. Flujo de Trabajo. Es una abstracción de un proceso de negocio, siendo este proceso de negocio conformado principalmente por reglas de negocio, actores, tareas y las dependencias entre estas tareas. Los estudios [41],[50],[53],[55],[61],[71] utilizan flujos de trabajo.
6. Planificación. Se refiere al uso de estrategias para definir una secuencia de actividades cuya finalidad es alcanzar un objetivo predefinido. Los estudios [27],[33],[36],[38],[39],[42],[43],[45],[52],[55],[58] muestran el uso de la técnica de planificación.
7. Modelo de Representación Interna. Es una forma para enlazar las dependencias entre los servicios web. Los estudios [19],[20],[22],[21],[33],[36],[37],[44],[51],[55],[64],[68] utilizan modelos de representación.
8. Resolución de Problemas. Se refiere a la búsqueda de solución a un problema dado dentro de un espacio de posibilidades. Los estudios [20],[29],[30],[59],[60],[65] muestran varios tipos de búsqueda. La diferencia entre la resolución de problemas y la planificación es en la



representación de estados, metas, acciones y la secuencia de acciones. Por ejemplo en la resolución de problemas los estados son entidades únicas mientras que en la planificación tienen representaciones estructuradas.

9. Razonamiento basado en Reglas. En este estudio [69], presenta un modelo de composabilidad para verificar si los servicios web candidatos a formar parte en una composición, realmente pueden interactuar entre ellos y si combinándolos otorgan un valor agregado. El modelo incluye un conjunto de reglas sintácticas y semánticas para comparar las propiedades de los servicios web.

Componente	Estudios	Conteo	%
Razonamiento sobre Ontologías	[21],[22],[23],[24],[25],[28],[31],[32],[34],[35],[37],[38],[40],[41],[42],[43],[44],[45],[46],[47],[48],[49],[50],[51],[52],[53],[54],[55],[56],[57],[63],[64],[65],[66],[67],[68],[69],[70],[71]	39	45.88
Razonamiento Lógico	[19],[23],[24],[26],[62],[66],[67]	7	8.24
Razonamiento Híbrido	[18]	1	1.18
Razonamiento Basado en Casos	[54]	1	1.18
Flujo de Trabajo	[41],[50],[53],[55],[61],[71]	6	7.06
Planificación.	[27],[33],[36],[38],[39],[42],[43],[45],[52],[55],[58]	11	12.94
Modelos de Representación Interna	[19],[20],[21],[22],[33],[36],[37],[44],[51],[55],[59],[64],[68]	13	15.29
Resolución de Problemas	[20],[29],[30],[59],[60],[65]	6	7.06
Razonamiento basado en reglas	[69]	1	1.18

**Tabla 6.16 Campo de aplicación**

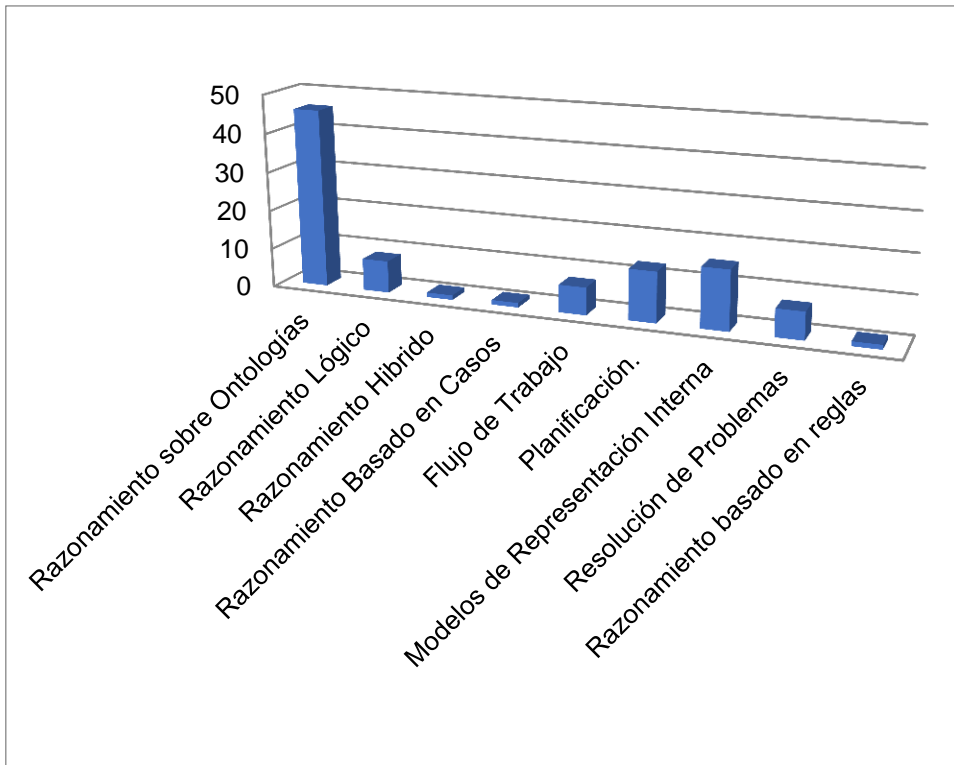


Figura 6.14 Campo de aplicación

## b) Lista de algoritmos o técnicas utilizadas en la arquitectura.

Para responder esta pregunta se hace referencia a los resultados mostrados en la tabla 6.16.

1. Razonamiento sobre Ontologías. En el razonamiento de ontologías se utilizan tres tipos de técnicas: razonamiento con lógica de primer orden, razonamiento con lógica descriptiva y razonamiento con reglas.
  - Razonamiento con lógica descriptiva. Los estudios [21],[22],[23],[24],[25],[28],[32],[37],[38],[40],[41],[42],[43],[44],[45],[46],[48],[49],[50],[51],[52],[53],[54],[57],[63],[64],[65],[66],[67],[68],[69],[71] utilizan algoritmos basado en lógica descriptiva.
  - Razonamiento con reglas. Los estudios [31],[34],[35],[47],[55],[56],[70] utilizan técnicas de lógica deductiva denominadas encadenamiento hacia adelante y encadenamiento hacia atrás. Se entiende por encadenamiento hacia atrás, cuando se parte de los estados metas y se tratan de cumplir las condiciones necesarias para llegar a ellos haciendo uso de ontologías y razonamiento semántico.

2. **Razonamiento Lógico.** Se refiere cuando los requerimientos y las descripciones de los servicios web existentes se traducen en axiomas.
  - **Formalismo de Conocimiento.** Los estudios [19],[23],[24],[66],[67] utilizan motores de composición de servicios basados en probadores de teoremas de lógica lineal .El artículo [19] usa además la técnica de deducción parcial que es un refinamiento de la lógica clásica.
  - **Formalismo de Acción.** Los estudios [26] utilizan Event Calculus. El estudio [62] utiliza un probador lógico de teoremas abductivos para lograr un plan de composición.
3. **Razonamiento Híbrido**
  - a. **Inferencia Híbridas.** Se refiere cuando se combina diferentes tipos de técnicas, tomando de las técnicas base sus mayores fortalezas. El estudio [18] presenta el algoritmo EHTN-DL, siendo este una combinación de la técnica de planeamiento de inteligencia artificial (HTN) con lógica descriptiva y el uso de la ontología de modelamiento de servicios web (WSMO).
4. **Razonamiento basado en casos.** Este tipo de razonamiento reutiliza y adapta soluciones pasadas exitosas a nuevos problemas [54]
5. **Flujo de Trabajo.**
  - a. **Orientado al Modelamiento.**
    - **MDA.** En los estudios [50],[53],[55] se utilizan diagramas UML .Los diagramas utilizados son : diagramas de clase, diagramas de actividad y diagramas de estados para modelar composiciones estáticas o dinámicas. También se encontró el uso de modelos CIM para modelar y reutilizar servicios web semánticos.
    - **MDD.** En el estudio [41], utiliza diagramas de actividad UML, donde cada actividad representa un servicio. En el estudio [61] se utiliza dos modelos: modelo independiente de la plataforma y modelo específico a la plataforma.
  - b. **No definido.** En los estudios [42],[71] se utiliza un flujo de trabajo personalizado y a partir de este se enlaza los servicios web semánticos. . En el estudio [71] se utiliza un lenguaje de modelamiento de servicios web compuestos (CWSL) que permite especificar la estructura de flujo de la composición con sus restricciones.
6. **Planificación.**
  - a. **Clásica.** Los estudios [33],[36],[38],[45] utilizan el lenguaje PDDL. El estudio [33] emplea una técnica de planeamiento en línea que se refiere a que los procesos de planificación, ejecución y monitoreo se efectúan concurrentemente. El lenguaje PDDL no permite representar acciones no determinísticas es decir acciones cuyos efectos no son predecibles o conocidas.

- b. Jerárquica. Los estudios [27],[39],[43],[52],[55] utilizan algoritmos de red jerárquica de tareas (HTN)
- c. Otros. Los estudios [42],[58] solo indica el uso de un método de planeamiento en forma general.

#### 7. Modelo de Representación Interna

Los estudios [19],[44],[55],[59],[64],[68] utilizan gráficos acíclicos dirigidos(DAG) para representar la composición, donde los nodos representan servicios y parámetros y los arcos representan dependencias. El estudio [68] presenta dos fases, en primer lugar se construye red de enlaces semánticos manejados de los servicio existentes (CLSN) y a partir de allí en respuesta a una solicitud de servicios se elabora un gráfico de dependencias a cíclico que permite la secuencia, selección y paralelismo .El estudio utiliza [20] utiliza gráficos And/Or para representar la composición de servicios. Los estudios [33],[36] utiliza el gráfico de planificación relajada RPG que provee la necesaria información heurística para la construcción de los planes , donde cada nivel del gráfico se asocia a un costo de métrica .El estudio [21] menciona el uso de un método de construcción de un gráfico en donde no debe haber una dependencia de un servicio así mismo .En el estudio [22] , menciona el uso de un lenguaje de descripción de composición de procesos basados en servicios web semánticos (CPSDL) .El estudio [37] se utiliza una matriz donde se relacionan los servicios web de acuerdo a un enlace formal llamado "enlace causal". El "enlace causal" es una dependencia lógica entre los parámetros de entrada y salida de diferentes servicios web. El estudio [51] se traduce el modelo de proceso OWL-S en una red Petri descrita en un archivo PNML.

#### 8. Resolución de problemas .En la resolución de problemas es necesario reducir los espacio de búsqueda debido a la gran cantidad de servicios disponibles y el tamaño de cada modelo de ontología .

- a. Búsqueda No Informada. El estudio [30] utiliza el algoritmo IDDFS. El estudio [59] utiliza el algoritmo BFS.
- b. Búsqueda Informada. Los estudio [20],[29],[65] utilizan funciones heurísticas. El estudio [20] utiliza el algoritmo heurístico AO\* que permite encontrar una solución gráfica con un costo mínimo, el estudio [65] utiliza el OWLS-Xplan. El Xplan es un planificador de búsqueda híbrido heurístico basado en FFplanner.
- c. Algoritmos BioInspirados.  
Algoritmo Evolutivo. El estudio [60] utiliza un algoritmo genético para evaluar los servicios web.

9. Razonamiento basado en reglas. El estudio [69] define descripciones de alto nivel de la composición deseada utilizando el lenguaje CSSL.

Técnica	Estudios	Conteo	%
Razonamiento con Lógica descriptiva	[21],[22],[23],[24],[25],[28],[32],[37],[38],[40],[41],[42],[43],[44],[45],[46],[48],[49],[50],[51],[52],[53],[54],[57],[63],[64],[65],[66],[67],[68],[69],[71]	32	37.65
Razonamiento con reglas	[31],[34],[35],[47],[55],[56],[70]	7	8.24
Formalismo de Conocimiento	[19],[23],[24],[66],[67]	5	5.88
Formalismo de Acción.	[26],[62]	2	2.35
Inferencia Híbridas.	[18]	1	1.18
Razonamiento Basado en Casos	[54]	1	1.18
MDA	[50],[53],[55]	3	3.53
MDD	[41],[61]	1	1.18
Modelamiento sin definir	[42],[71]	2	2.35
Planificación Clásica	[33],[36],[38],[45]	4	4.71
Planificación Jerárquica	[27],[39],[43],[52],[55]	5	5.88
Planificación sin definir	[42],[58]	2	2.35
Modelo de Representación	[19],[20],[21],[22],[33],[36],[37],[44],[51],[55],[59],[64],[68]	13	15.29
Búsqueda no Informada	[30],[59]	2	2.35
Búsqueda informada	[20],[29],[65]	3	3.53
Algoritmo evolutivo	[60]	1	1.18
Razonamiento basado en reglas	[69]	1	1.18

Tabla 6.17 Lista de técnicas

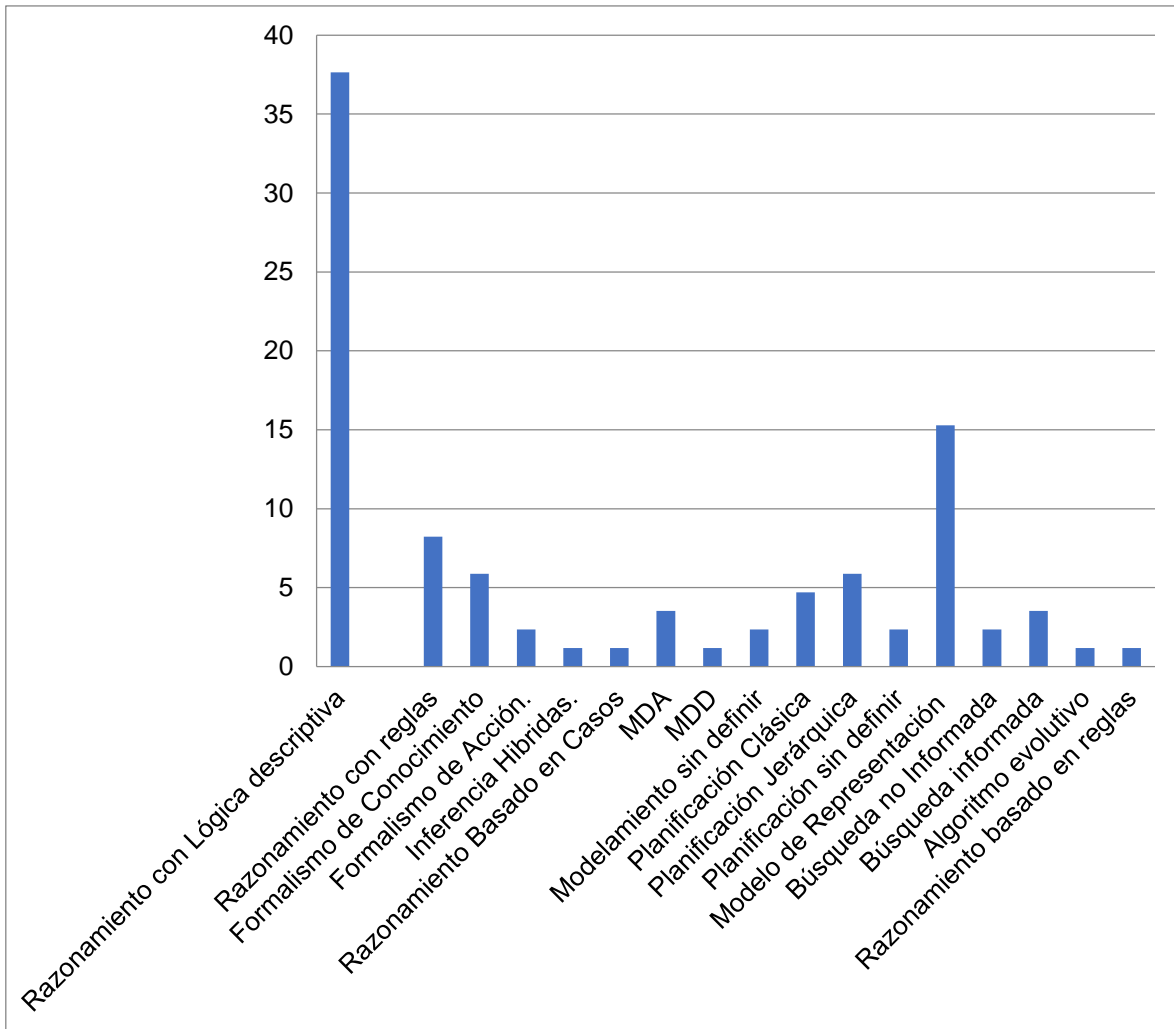


Figura 6.15 Técnicas utilizadas en la arquitectura

#### 6.2.4.1 Análisis y Discusión

Se puede sintetizar en cuatro categorías las técnicas encontradas en la tabla 6.17 las cuales son: Inferencia semántica, inteligencia artificial, inferencia lógica, flujo de trabajo, su utilización depende de cuan bien este definido la solicitud del usuario. Si un usuario requiere que un sistema le vaya guiando gradualmente hasta definir completamente su requerimiento, en este caso sería recomendable un entorno semi-automizado donde se combinan técnicas orientadas al flujo combinadas con inferencias semánticas , técnicas de inteligencia artificial y/o técnicas de inferencia lógica , caso contrario si un usuario tiene un requerimiento bien definido , sería recomendable un entorno automatizado en donde se utilizan técnicas de inferencias semánticas , técnicas de

inteligencia artificial y/o técnicas de inferencia lógica . Por lo tanto para abordar el problema de composición es necesario utilizar soluciones híbridas tomando las fortalezas de cada una de ellas Es necesario mencionar que se entiende por orientado a flujo a lo que especifica el usuario, por ejemplo una solución automatizada que emplea la inferencia semántica podría utilizar internamente diagramas acíclicos para representar las relaciones entre los servicios web semánticos no puede definirse como orientado al flujo.

Un importante aporte de los estudios [62], [69] es que proponen el descubrimiento y composición de servicios compuesto a diferencia del resto de estudios que solo proponen el descubrimiento y composición de servicios web atómicos.



## 6.2.5 ¿Cómo utilizar la inteligencia artificial en una arquitectura de servicios web semánticos compuestos?

Para responder esta pregunta se utiliza los resultados de las tablas 6.18 y 6.19 y de la tabla 0.3 del apéndice, además de los hallazgos encontrados en la búsqueda secundaria.

a) Rol de la inteligencia artificial.

1. Razonamiento. En este campo se considera el razonamiento basado en casos [54]. Este tipo de técnica se basa en la premisa que los nuevos problemas son similares a los problemas encontrados anteriormente, por tal motivo utiliza soluciones pasadas directamente o adaptándola mediante un ajuste.
2. Planificación. Se considera el problema de la composición de servicios web como un problema de planificación en Inteligencia Artificial. Todo plan presenta los siguientes componentes: El estado objetivo que se pretende alcanzar ( relación con los requerimientos del usuario ) , la acción o la secuencia de pasos ir de un estado a otro ( relación con los servicios atómicos dentro de una composición de servicios web ) , la condición previa ( las restricciones , precondiciones ) , y los efectos de cada acción ( salida de una composición). Los estudios [27],[33],[36],[38],[39],[42],[43],[45],[52],[55],[58] muestran el uso de la técnica de planificación. Los estudios [33],[36] proponen un modelo de planificación reactiva concurrente con el proceso de ejecución tomando en cuenta la restricción del tiempo , siendo el plan construido , refinado y ejecutado en forma incremental en tiempo real para que los planes puedan ser ejecutados tomando en cuenta un entorno indeterminado.
3. Procesamiento de Lenguaje Natural. Se refiere a las técnicas que permiten a las computadoras tener capacidad de entender, interpretar y manipular el lenguaje humano. Los estudios [54],[58],[64] utilizan este procesamiento para interpretar los requerimientos de los usuarios.
4. Sistemas Multiagente. Se refiere al estudio de la cooperación entre los diferentes tipos de agentes para lograr una adecuada composición de servicios web. Los estudios [25],[32],[58] utilizan agentes de captura de requerimientos ,en el estudio[25] se le denomina agente cliente , en el estudio [32] se les denominan agente solicitante y agente intermediario, en el estudio[58] se le denomina agente manejador de solicitudes. Los estudios [25],[32],[46],[65],[57],[58],[61] usan agentes para el descubrimiento del servicio , en el estudio



[32] se le denomina agente de ontología , en los estudios [25],[65],[58] se le denominan agente de descubrimiento de servicio ,en el estudio[25] se le denomina agente de regla de inferencia y agente de ontología de dominio , en los estudios [46] se le denomina motor de agente BDI , en los estudios [65],[57] se les denomina agentes de coincidencia semántica y en el estudio [61] se le denomina agente WSMX. Los estudios [25],[32],[36],[46],[57],[58],[60],[61],[65] utilizan agentes para la composición de servicios , en el estudio[25] se le denomina agente de regla de inferencia , en el estudio [32] utiliza un agente de ontología , en los estudios [46] se le denomina motor de agente BDI ,en los estudios [65],[57],[58],[61] se les denomina agente de composición de servicio y en los estudios [36],[60] se le denomina agente de planeamiento. El estudio [58] utilizan agentes para la evaluación de la composición. Los estudios [36],[58],[65] utilizan agentes para la ejecución , en el estudio[65] se le denomina agente de planeamiento .El estudio [24] utilizan agentes de monitoreo para tener valores precisos y en tiempo de ejecución de las propiedades de calidad. Los estudios [32],[60] utilizan agentes de reconfiguración, en el estudio [32] se le denomina agente proxy y en el estudio [60] se le denomina agente manejador de excepciones cuando ocurre un error durante el proceso de ejecución .

5. Resolución de Problemas. Se refiere al proceso en el cual se elige una secuencia de acciones entre varios estados posibles para alcanzar un estado deseado. Los estudios [20],[29],[65] muestran métodos de búsqueda heurística . El estudio [60] tiene un algoritmo genético.

Rol	Estudios	Conteo	%
Razonamiento	[54]	1	3.45
Planificación	[27],[33],[36],[38],[39],[42],[43],[45],[52],[55],[58]	11	37.93
Procesamiento de Lenguaje Natural..	[54],[58],[64]	3	10.34
Sistemas Multiagente	[24],[25],[32],[36],[46],[57],[58],[60],[61],[65]	10	34.48
Búsqueda	[20],[29],[60],[65]	4	13.79

**Tabla 6.18 Rol de la Inteligencia Artificial**

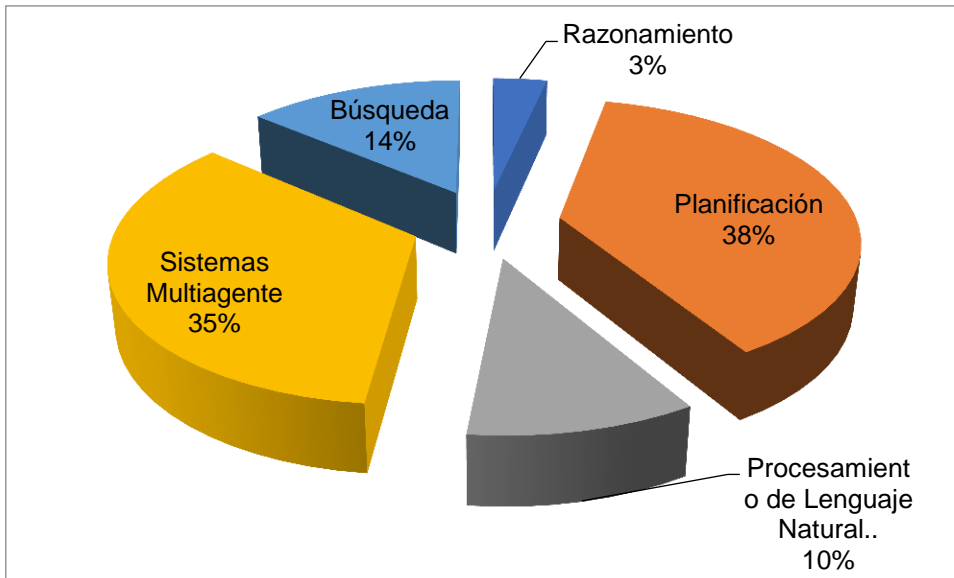


Figura 6.16 Rol de la inteligencia artificial

b) Etapa en el que interviene.

1. Fase Especificación. Esta fase corresponde a la obtención de información semántica a partir de los servicios web y de la solicitud del usuario. En los estudios [25],[32] se utilizan agentes en la captura de requerimientos . En los estudios [54],[58],[64] se utilizan técnicas de procesamiento de lenguaje natural para capturar los requerimientos a partir de la voz del usuario.
2. Fase Descubrimiento. Esta fase corresponde a la búsqueda de servicios web atómicos o compuestos que puedan satisfacer el requerimiento del solicitante. En los estudios [20],[25],[29],[32],[46],[65],[57],[58],[61] se utilizan agente de descubrimiento y el uso de técnicas de búsquedas heurísticas.
1. Fase Composición. Esta fase corresponde a la elaboración de una solución que satisfaga un requerimiento complejo. Los estudios [25],[27],[32],[33],[36],[38],[39],[42],[43],[45],[46],[52],[55],[57],[58],[60],[61],[65] utilizan agentes de planeamiento , la técnica de planificación de la inteligencia artificial y el razonamiento basado en casos.
2. Fase Evaluación. Esta fase corresponde a la selección de la mejor solución de entre varias alternativas tomando en cuenta los parámetros no funcionales Los estudios [58],[60] utilizan agentes de evaluación y el uso de un algoritmo genético.
3. Fase Ejecución. En esta fase se ejecuta la mejor solución. Los estudios [36],[58],[65]

muestran agentes de ejecución.

4. Fase Monitoreo dinámico. El estudio [24] solo efectúa la acción de monitoreo .Los estudios [32],[60] se muestran agentes que efectúan monitoreo y reconfiguración.

Fase	Estudios	Conteo	%
Especificación	[25],[32],[54],[58],[64]	5	13.16
Descubrimiento	[20],[25],[29],[32],[46],[65],[57],[58],[61]	9	23.68
Composición	[25],[27],[32],[33],[36],[38],[39],[42],[43],[45],[46],[52],[54],[55],[57],[58],[60],[61],[65]	18	47.37
Evaluación	[58],[60]	2	5.26
Ejecución	[36],[58],[65]	3	7.89
Monitoreo Dinámico	[24],[32],[60]	1	2.63

Tabla 6.19 Etapa en la que interviene

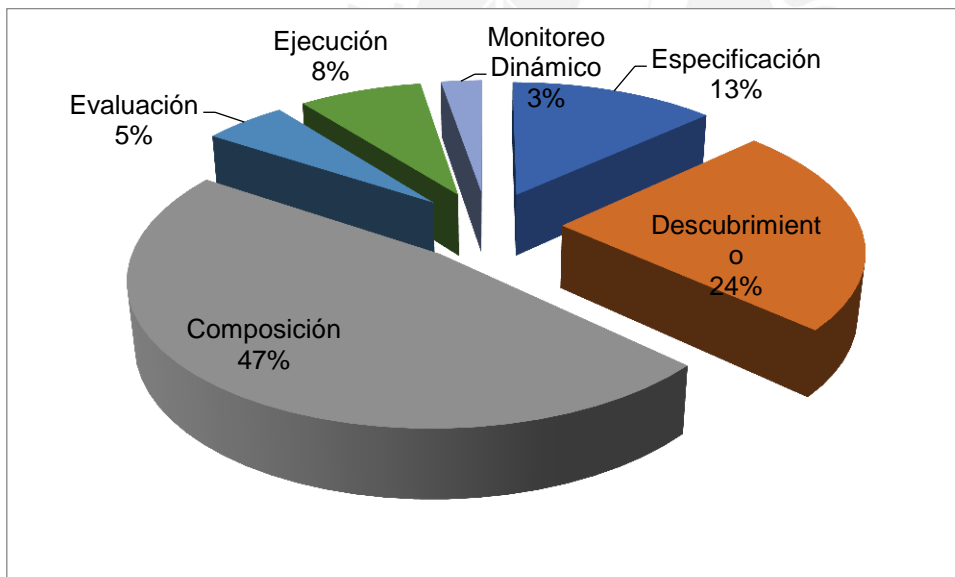


Figura 6.17 Etapa en la intervenció

### 6.2.5.1 Análisis y Discusión

El rol principal de la inteligencia artificial es para facilitar la composición automatizada de los servicios web, esto se logra a través de técnicas de planificación, resolución de problemas y de agentes inteligentes capaces de trabajar conjuntamente para construir un plan de composición. Las técnicas de procesamiento de lenguaje natural se usan principalmente para convertir los mensajes de voz en una serie de conceptos a las cuales se pueden encontrar un nivel de correspondencia con las ontologías utilizadas por la arquitectura. Es necesario mencionar que pocos artículos mencionan planes de control y recuperación en tiempo de ejecución para asegurar el correcto funcionamiento de la composición, asimismo hay pocos estudios que tratan la generación de planes con contingencias, por lo tanto el manejo del no determinismo no está cubierto en las fases de ejecución y composición. Esto es crítico debido a que el mundo real es un entorno no determinista, debido a que no existe un único camino predecible para un conjunto de acciones.



# Capítulo 7.

## Conclusiones y Trabajos Futuros

En el presente trabajo de investigación, se encontraron trabajos colaborativos de investigadores provenientes de varias instituciones educativas, siendo este tipo de trabajo llevado casi en su totalidad por países de Europa, a su vez la mayoría de artículos por país que satisficieron los criterios de inclusión y exclusión y los criterios fueron los artículos provenientes de China. En los campos de aplicación en donde se busca implementar la arquitectura, la mayoría tienden a ser concretas implementadas en prototipos.

En respuesta a los módulos de arquitectura que soporta la composición de servicios web semánticos, se encontró que todas estaban compuestas por una arquitectura de n-capas, y el estilo arquitectural utilizada era SOA (arquitectura orientada a servicios) y P2P, además las arquitecturas encontradas, tienden hacia un esquema automatizado pero muy pocas son dinámicas porque las composiciones de servicios web no pueden ser recompuestas en tiempo de ejecución en respuesta a un evento dado un problema en el funcionamiento de la composición.

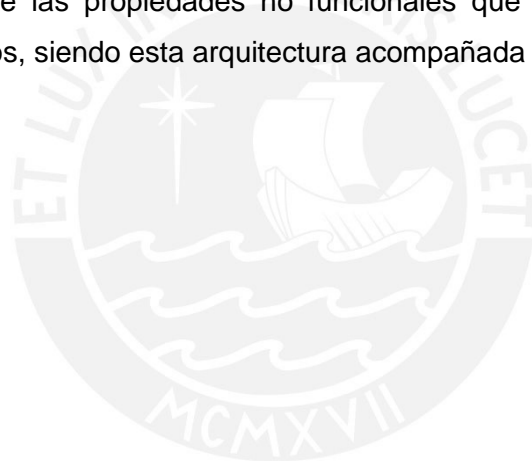
En respuesta a las ontologías utilizadas, se encontró que la mayoría utiliza lenguajes de ontologías estandarizadas ampliamente aceptadas por las comunidades en todas sus variantes, otras usan extensiones de estas ontologías, cuando se necesita agregar ciertos conceptos como por ejemplo el conocimiento del contexto que no son soportados por estas ontologías estandarizadas. Sin embargo se encontró que todas las arquitecturas siempre hacen referencia en su mayoría en ontologías de dominio y de servicio, esto se puede explicar ya que es a través de la ontología es que se logra la automatización en la composición de los servicios semánticos.

En respuesta a las capacidades de seguridad que deben ser soportadas por la arquitectura, se encontró muy pocos artículos que trataban puntualmente el tema de seguridad, los estudios encontrados incluían el tema de seguridad dentro de propiedades no funcionales o de calidad, siendo las propiedades funcionales definidas en una ontología, almacenadas en un repositorio y utilizadas fundamentalmente con un propósito de seleccionar la mejor composición de servicios web semánticos que satisfaga los requerimientos del solicitante, muy pocos estudios utilizan algoritmos de calidad.

En respuesta a las técnicas de composición utilizadas en las arquitecturas, se encontró que la mayoría tiende a utilizar técnicas de Inteligencia artificial, de inferencia semántica y de inferencia lógica esto es debido a que estas técnicas responden mejor a una composición automática. Se encontraron muy pocas propuestas utilizando MDA, las cuales no es una propuesta MDA pura si no que es combinada con otras técnicas arriba mencionadas.

En respuesta al rol de la inteligencia artificial, si principal función es lograr una composición automatizada, se encontró el uso de la técnica de planeamiento de la inteligencia artificial, técnicas de búsqueda utilizando algoritmos heurísticos, agentes y técnicas de procesamiento de lenguaje natural.

Como trabajo futuro, se tiene varios temas de investigación, un tema interesante sería desarrollar una arquitectura que soporte la planificación no determinada, reactiva, escalable y con un buen rendimiento sustentada en pruebas no funcionales. Otro tema interesante de abordar sería una arquitectura que soporte las propiedades no funcionales que garantice la mejor composición de servicios web semánticos, siendo esta arquitectura acompañada de un algoritmo de calidad.



# Apéndice

## I. Lista de Módulos

ID	Módulo	Patrón	Nivel
[18]	<ul style="list-style-type: none"> <li>• El Planeador EHTN-DL</li> <li>• El Traductor AIMO</li> <li>• Módulo WSMX</li> <li>• El Razonador WSML-DL</li> </ul>	N capas	automático
[19]	<ul style="list-style-type: none"> <li>• La capa de descomposición de problemas</li> <li>• El motor de composición de servicios</li> <li>• La capa de asignación de tareas</li> <li>• El motor de flujo de trabajo</li> <li>• La capa de comunicación</li> </ul>	N capas	automático
[20]	<ul style="list-style-type: none"> <li>• La ontología de conocimiento</li> <li>• La Plantilla de tareas</li> <li>• El planeador</li> <li>• El mediador</li> <li>• Los Servicios Web</li> </ul>	N capas	automático
[21]	<ul style="list-style-type: none"> <li>• El centro de registro</li> <li>• La Base de datos QoS</li> <li>• La Base de Conocimiento</li> <li>• El motor de búsqueda</li> <li>• El motor de composición</li> <li>• El motor de ejecución</li> <li>• La Base de datos Web</li> <li>• El combinador</li> </ul>	N capas	automático
[22]	<ul style="list-style-type: none"> <li>• Recolección y clasificación de servicios</li> <li>• Anotador OWL-S</li> <li>• Despliegue de servicios en P2P</li> <li>• Motor de descubrimiento de servicios</li> <li>• Selección y evaluación de servicios</li> <li>• Composición de servicios</li> <li>• Ejecución y validación de servicios</li> </ul>	N capas	semiautomático
[23]	<ul style="list-style-type: none"> <li>• La interfaz del Usuario</li> <li>• El traductor</li> </ul>	N capas	automático

	<ul style="list-style-type: none"> <li>• El examinador del teorema de lógica lineal</li> <li>• El adaptador</li> <li>• El razonador semántico</li> <li>• La base de ontología</li> </ul>		
[24]	<ul style="list-style-type: none"> <li>• La unidad de control</li> <li>• El administrador de solicitudes</li> <li>• El administrador de ontologías</li> <li>• El administrador de la calidad</li> <li>• El administrador central</li> </ul>	N capas	automático
[25]	<ul style="list-style-type: none"> <li>• Herramienta grafica de administración de administración de sistemas.</li> <li>• Editor visual de servicios web semánticos</li> <li>• ISCFAT</li> <li>• Compilador de Servicio web</li> <li>• Buscador de Servicio Web</li> <li>• Motor de coincidencia OWL-S</li> <li>• Administrador de ejecución de servicios web</li> <li>• Administrador de recursos</li> <li>• Adaptador de mensajes</li> </ul>	N capas	Manual/ Semiautomático/ Automático
[26]	<ul style="list-style-type: none"> <li>• El planeador de Flujo de Trabajo <ul style="list-style-type: none"> <li>▪ La pila de Metas</li> <li>▪ La memoria de trabajo</li> <li>▪ Interpretador</li> </ul> </li> <li>• El modelado de dominio <ul style="list-style-type: none"> <li>▪ La ontología de Dominio</li> <li>▪ La ontología de tareas</li> <li>▪ La ontología de Metas</li> </ul> </li> <li>• El bróker de Servicios</li> <li>• El motor del flujo de trabajo publicado</li> <li>• La interfaz de usuario</li> <li>• El registro de servicios semánticos</li> <li>• El repositorio local de flujos semánticos</li> </ul>	N capas	automático
[27]	<ul style="list-style-type: none"> <li>• El monitor</li> <li>• El planeador HTN</li> <li>• El traductor de OWL-S a HTN</li> <li>• El convertidor de HTN a OWL-S</li> <li>• El ejecutor OWL-S</li> <li>• El dominio de planeamiento HTN</li> </ul>	N capas	automático
[28]	<ul style="list-style-type: none"> <li>• La interfaz de usuario</li> <li>• El módulo de descripción semántica de los</li> </ul>	N capas	automático



	<p>servicios web</p> <ul style="list-style-type: none"> <li>• El módulo de descubrimiento automático de servicios web</li> <li>• El módulo de composición de servicios web semánticos <ul style="list-style-type: none"> <li>▪ Módulo de alineamiento de terminología</li> <li>▪ Módulo de alineamiento extrínseco</li> <li>▪ Módulo de planeamiento</li> <li>▪ Módulo de ejecución</li> </ul> </li> </ul>		
[29]	<ul style="list-style-type: none"> <li>• La Interface de servicios web</li> <li>• El analizador de entradas basado en SAX</li> <li>• El planificador de la estrategia</li> <li>• La base de conocimiento</li> <li>• El registro de servicios</li> <li>• El escritor de salida basada en SAX</li> <li>• El compositor Heurístico</li> <li>• El compositor IDDFS</li> <li>• El compositor de algoritmo genético</li> </ul>	N capas	automático
[30]	<ul style="list-style-type: none"> <li>• La Interface de servicios web</li> <li>• El analizador de entradas basado en SAX</li> <li>• El planificador de la estrategia</li> <li>• La base de conocimiento</li> <li>• El registro de servicios</li> <li>• El escritor de salida basada en SAX</li> <li>• El compositor Heurístico</li> <li>• El compositor IDDFD</li> <li>• El compositor de algoritmo genético</li> </ul>	N capas	automático
[31]	<ul style="list-style-type: none"> <li>• EL repositorio de servicios OWL-S</li> <li>• El api OWL-S</li> <li>• El motor de composición</li> <li>• EL motor de inferencia</li> <li>• El emparejador OWL-S</li> </ul>	N capas	automático
[32]	<ul style="list-style-type: none"> <li>• El agente corredor</li> <li>• El agente de requerimiento</li> <li>• EL agente de ontología</li> <li>• El agente proveedor</li> <li>• El agente proxy</li> <li>• El administrador de la reconfiguración</li> <li>• El administrador de decisiones</li> </ul>	N capas	automático
[33]	<ul style="list-style-type: none"> <li>• El traductor de OWLS – hacia una especificación</li> </ul>	N capas	automático

	<p>PDDL</p> <ul style="list-style-type: none"> <li>• El modelador del entorno</li> <li>• El planificador</li> <li>• El ejecutor</li> </ul>		
[34]	<ul style="list-style-type: none"> <li>• El repositorio OWL-S.</li> <li>• La librería OWL-S API</li> <li>• El motor de composición</li> <li>• El sistema de descubrimiento de servicios web semánticos</li> </ul>	N capas	automático
[35]	<ul style="list-style-type: none"> <li>• El manejador de consultas</li> <li>• El manejador de la continuación de la fase</li> <li>• El emparejador semántico</li> <li>• El administrador de la memoria cache</li> <li>• El escuchador de redes</li> </ul>	N capas	automático
[36]	<ul style="list-style-type: none"> <li>• El traductor de la especificación de OWL-S a PDDL</li> <li>• Un modelador del entorno</li> <li>• El módulo planificador</li> <li>• El módulo de ejecución</li> </ul>	N capas	automático
[37]	<ul style="list-style-type: none"> <li>• El modulo del descubrimiento y selección de servicios web</li> <li>• El módulo de composición a nivel funcional <ul style="list-style-type: none"> <li>▪ El compositor de servicios web</li> </ul> </li> <li>• El módulo de composición a nivel de proceso</li> </ul>	N capas	automático
[38]	<ul style="list-style-type: none"> <li>• El orquestador de Servicio</li> <li>• El planificador de la nube</li> <li>• El solucionador de problemas</li> <li>• El monitor de servicio</li> <li>• La ontología de Servicio</li> </ul>	N capas	automático
[39]	<ul style="list-style-type: none"> <li>• La interface del usuario final</li> <li>• La interface administrativa</li> <li>• El modulo descubridor</li> <li>• El modulo traductor</li> <li>• El modulo ejecutor</li> <li>• La ontología de dominio y de servicio</li> <li>• El repositorio de servicios</li> <li>• El compositor</li> <li>• El evaluador</li> </ul>	N capas	Manual/ Semiautomático/ Automático

[40]	<ul style="list-style-type: none"> <li>• EL módulo de interface del usuario</li> <li>• El módulo de descubrimiento automático de servicios web semánticos</li> <li>• El módulo de planificación</li> <li>• El módulo de optimización</li> </ul>	N capas	automático
[41]	<ul style="list-style-type: none"> <li>• El módulo de modelamiento de servicio</li> <li>• El módulo de la composición de servicio</li> <li>• El modulo del despliegue del servicio</li> <li>• El motor de arquitectura de servicio</li> </ul>	N capas	semiautomático
[42]	<ul style="list-style-type: none"> <li>• El proceso del modelamiento del entorno</li> <li>• El repositorio de artefactos ejecutables</li> <li>• El componente descubridor</li> <li>• El componente de mediación</li> <li>• El componente de composición</li> </ul>	N capas	semiautomático
[43]	<ul style="list-style-type: none"> <li>• EL motor de ejecución</li> <li>• El selector de servicios semánticos</li> <li>• El motor de transformación</li> <li>• El planificador</li> <li>• El repositorio de servicios semánticos</li> </ul>	N capas	automático
[44]	<ul style="list-style-type: none"> <li>• El triple generador</li> <li>• El lector de consulta</li> <li>• El procesador de consultas de descubrimiento</li> <li>• El motor de composición</li> <li>• El generador de salida</li> </ul>	N capas	automático
[45]	<ul style="list-style-type: none"> <li>• El modelador del contexto de la ontología</li> <li>• El módulo de descubrimiento y razonamiento</li> <li>• El planificador de la composición. <ul style="list-style-type: none"> <li>▪ El Traductor de OWL-S a PDDL</li> <li>▪ El planificador</li> <li>▪ El traductor de PDDL a OWL-S</li> <li>▪ La herramienta de adaptación</li> <li>▪ El repositorio de adaptación</li> </ul> </li> </ul>	N capas	automático
[46]	<ul style="list-style-type: none"> <li>• Ontología de Aplicación.</li> <li>• Motor de agente BDI basado en ontología <ul style="list-style-type: none"> <li>▪ Motor de metas</li> <li>▪ Motor de creencias</li> <li>▪ Motor de Flujo de trabajo</li> <li>▪ Motor de acción</li> </ul> </li> <li>• Ontología operacional</li> </ul>	N capas	automático

	<ul style="list-style-type: none"> <li>• Plataforma Jadex</li> </ul>		
[47]	<ul style="list-style-type: none"> <li>• El administrador semántico</li> <li>• El motor de composición local</li> <li>• El módulo de composición P2P</li> </ul>	N capas	automático
[48]	<ul style="list-style-type: none"> <li>• La ontología IRPG</li> <li>• Los servicios de aplicación</li> <li>• Los servicios de flujo de trabajo</li> <li>• El registro de servicios</li> </ul>	N capas	automático
[49]	<ul style="list-style-type: none"> <li>• El servicio administrador de la ontología</li> <li>• El servicio administrador de la composición</li> <li>• El administrador de transcripción inteligente</li> </ul>	N capas	automático
[50]	<ul style="list-style-type: none"> <li>• El centro de registro</li> <li>• La base de conocimiento</li> <li>• La base de datos QoS</li> <li>• Motor de búsqueda</li> <li>• Motor de composición</li> <li>• Motor de ejecución</li> </ul>	N capas	semiautomático
[51]	<ul style="list-style-type: none"> <li>• El motor de búsqueda <ul style="list-style-type: none"> <li>▪ El analizador funcional</li> <li>▪ El analizador de comportamiento</li> </ul> </li> <li>• El registro local</li> <li>• El administrador de registro</li> <li>• El administrador de cuentas</li> <li>• El convertidor de OWL-S a PNML</li> <li>• El generador de Hipergramas</li> <li>• El módulo SEMFIT</li> <li>• El explorador de servicios</li> </ul>	N capas	automático
[52]	<ul style="list-style-type: none"> <li>• El servicio compositor</li> <li>• El servicio analizador del modelo</li> <li>• El servicio traductor del esquema</li> <li>• El servicio de adaptación monitoreo y ejecución</li> </ul>	N capas	automático
[53]	<ul style="list-style-type: none"> <li>• La capa CIM(descubrimiento razonamiento, reutilización , readaptación )</li> <li>• La capa PIM(emparejamiento)</li> <li>• La capa PSM(Ejecución)</li> </ul>	N capas	semiautomático
[54]	<ul style="list-style-type: none"> <li>• La capa de extracción de metas del usuario.</li> <li>• La capa de emparejamiento de casos</li> </ul>	N capas	semiautomático

	<p>semánticos.</p> <ul style="list-style-type: none"> <li>• La capa de revisión y adaptación de casos.</li> <li>• La ejecución de los casos.</li> </ul>		
[55]	<ul style="list-style-type: none"> <li>• El proxy de contexto</li> <li>• El repositorio de contextos</li> <li>• La especificación del contexto</li> <li>• La librería de planes</li> <li>• El planificador global</li> <li>• EL módulo de optimización local</li> <li>• El motor de ejecución</li> <li>• El registro de servicios OWL-S</li> </ul>	N capas	automático
[56]	<ul style="list-style-type: none"> <li>• El motor de inferencias</li> <li>• El compositor</li> <li>• La Base de conocimiento</li> <li>• El Traductor DAML-S</li> <li>• El Ejecutor DAML-S</li> </ul>	N capas	semiautomático
[57]	<ul style="list-style-type: none"> <li>• El agente de emparejamiento <ul style="list-style-type: none"> <li>▪ El motor de emparejamiento semántico</li> <li>▪ El analizador DAML-S</li> </ul> </li> <li>• El servidor de razonamiento ICS <ul style="list-style-type: none"> <li>▪ El razonador de la lógica de descripción</li> </ul> </li> </ul>	N capas	semiautomático
[58]	<ul style="list-style-type: none"> <li>• Capa de comunicación</li> <li>• Capa de composición <ul style="list-style-type: none"> <li>▪ El agente manejador de solicitudes</li> <li>▪ El agente descubridor</li> <li>▪ El agente planificador</li> <li>▪ El agente evaluador</li> <li>▪ El agente ejecutor</li> </ul> </li> <li>• Capa de ontología</li> <li>• Capa de sistemas heredados</li> </ul>	N capas	automático
[59]	<ul style="list-style-type: none"> <li>• Servicio de Ontología</li> <li>• El monitor de servicio</li> <li>• El motor de Orquestación</li> </ul>	N capas	automático
[60]	<ul style="list-style-type: none"> <li>• La unidad de análisis de requerimiento</li> <li>• El agente planificador</li> <li>• El agente del emparejamiento semántico</li> <li>• La unidad de ejecución de proceso</li> </ul>	N capas	automático

	<ul style="list-style-type: none"> <li>• El agente manejador de excepciones</li> </ul>		
[61]	<ul style="list-style-type: none"> <li>• Módulo de ejecución Semántica</li> <li>• Módulo de descubrimiento</li> <li>• Módulo de composición</li> <li>• Módulo de selección y clasificación</li> <li>• Módulo de mediación de datos</li> <li>• Módulo de coreografía</li> <li>• Grounding</li> <li>• Invocador</li> <li>• Interface administrador de recursos</li> </ul>	N capas	automático
[62]	<ul style="list-style-type: none"> <li>• El módulo de descubrimiento de servicio <ul style="list-style-type: none"> <li>▪ El algoritmo de emparejamiento de procesos atómicos</li> <li>▪ Repositorio de servicio</li> <li>▪ Ontología de servicio</li> <li>▪ Lista de Servicios</li> </ul> </li> <li>• El módulo de composición y ejecución <ul style="list-style-type: none"> <li>▪ Generador de axiomas</li> <li>▪ Motor de inferencia</li> <li>▪ Plan</li> </ul> </li> </ul>	N capas	automático
[63]	<ul style="list-style-type: none"> <li>• Analizador Semántico</li> <li>• Módulo de descubrimiento</li> <li>• Módulo de orquestación</li> <li>• Módulo de entrega</li> <li>• Repositorio semántico</li> <li>• Repositorio de servicios</li> <li>• Repositorio de servicios semánticos Internet/Intranet</li> </ul>	N capas	automático
[64]	<ul style="list-style-type: none"> <li>• El interpretador de solicitudes</li> <li>• El catálogo de servicios semánticos</li> <li>• El motor generador de servicios</li> <li>• La base de conocimiento</li> </ul>	N capas	automático
[65]	<ul style="list-style-type: none"> <li>• La capa de red</li> <li>• La capa de coordinación de servicio</li> <li>• El subsistema de contexto</li> <li>• El subsistema de privacidad y seguridad</li> </ul>	N capas	automático

[66]	<ul style="list-style-type: none"> <li>• El traductor</li> <li>• La interface de usuario</li> <li>• El probador del teorema LL</li> <li>• El razonador semántico</li> <li>• El adaptador</li> </ul>	N capas	automático
[67]	<ul style="list-style-type: none"> <li>• Analizador DAML-S</li> <li>• Interfaz de usuario</li> <li>• Coordinador</li> <li>• El planificador de lógica lineal</li> <li>• Razonador semántico</li> <li>• La base de ontología</li> </ul>	N capas	automático
[68]	<ul style="list-style-type: none"> <li>• El Repositorio RDF</li> <li>• El Modulo de selección y descubrimiento de servicio</li> <li>• El razonador semántico</li> <li>• El módulo de composición flexible</li> <li>• El ejecutor de servicio</li> </ul>	N capas	automático
[69]	<ul style="list-style-type: none"> <li>• Generador WSFL</li> <li>• Generador XLANG</li> <li>• Manejador de solicitud</li> <li>• Administrador de servicios compuestos</li> <li>• El localizador de servicio</li> <li>• El controlador</li> <li>• Matchmaker</li> <li>• El Repositorio de plantillas</li> <li>• El optimizador basado en QoS</li> </ul>	N capas	automático
[70]	<ul style="list-style-type: none"> <li>• Ontología de Dominio</li> <li>• Modelos BR(reglas de negocio)</li> <li>• Ontología SWS ( servicios web semánticos)</li> <li>• Motor de inferencia</li> <li>• Compositor</li> <li>• Ejecución de servicio</li> </ul>	N capas	automático
[71]	<ul style="list-style-type: none"> <li>• Interfaz de usuario</li> <li>• El modelador de servicios compuestos</li> <li>• El planificador de la composición <ul style="list-style-type: none"> <li>▪ El motor de selección y emparejamiento de servicios.</li> <li>▪ El motor de consulta de servicio</li> <li>▪ El motor de inferencia semántica</li> <li>▪ El generador de plan</li> </ul> </li> </ul>	N capas	semiautomático

	<ul style="list-style-type: none"> <li>▪ El solucionador de restricciones</li> <li>• El registro UDDI</li> <li>• El repositorio de ontología</li> <li>• El repositorio de contexto</li> </ul>		
--	---	--	--

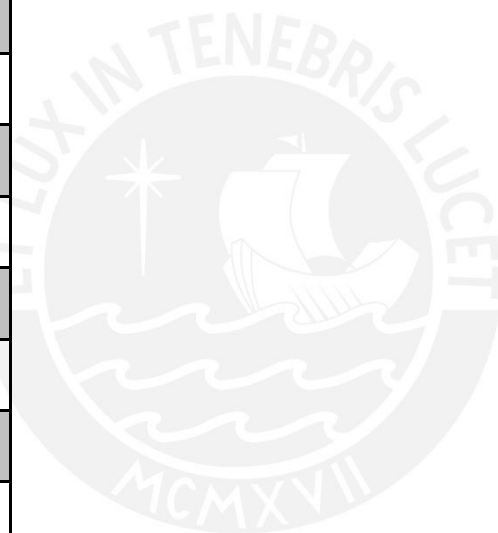
**Tabla 0.1 Lista de módulos de la arquitectura**



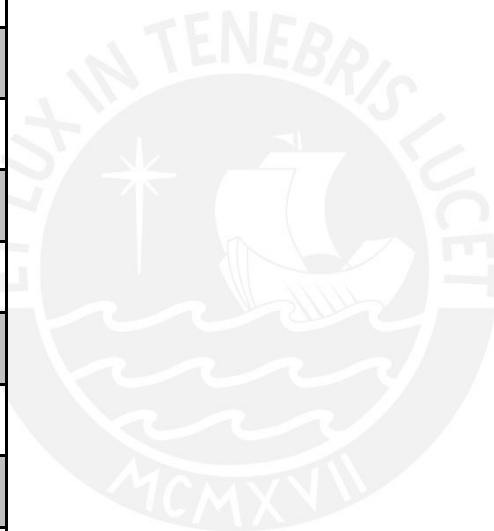


## II. Lista de Ontologías

Artículo	Lenguaje
[18]	WSMO
[19]	OWL ,OWL-S
[20]	Sin definir
[21]	OWL ,OWL-S
[22]	OWL ,OWL-S
[23]	DAML,DAML-S
[24]	Sin definir
[25]	ISCF Task
[26]	OWL,OWL-S
[27]	OWL,OWL-S
[28]	OWL,OWL-S
[29]	OWL,OWL-S
[30]	OWL,OWL-S
[31]	OWL,OWL-S
[32]	OWL,OWL-S
[33]	OWL,OWL-S
[34]	OWL,OWL-S
[35]	OWL,OWL-S
[36]	OWL,OWL-S
[37]	OWL,OWL-S WSMO
[38]	OWL,OWL-S

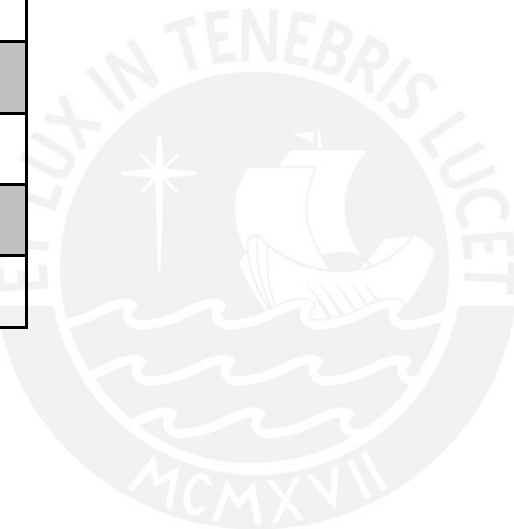


[39]	WSMO
[40]	OWL,OWL-S
[41]	WSMO,WSPO
[42]	Sin definir
[43]	OWL,OWL-S, WSMO
[44]	OWL,OWL-S
[45]	OWL,OWL-S OWL-SC
[46]	OWL,OWL-S
[47]	OWL,OWL-S
[48]	iRPG
[49]	OWL,OWL-S
[50]	OWL,OWL-S
[51]	OWL,OWL-S
[52]	OWL,OWL-S
[53]	OWL,OWL-S, OWL-DL , WSMO
[54]	CBROnto.
[55]	OWL-SC OWL,OWL-S
[56]	OWL,OWL-S
[57]	DAML,DAML-S
[58]	Sin definir
[59]	OWL,OWL-S



	SAWSDL
[60]	OWL,OWL-S
[61]	WSMO
[62]	OWL,OWL-S
[63]	OWL,OWL-S
[64]	OWL,OWL-S
[65]	OWL,OWL-S
[66]	DAML ,DAML-S
[67]	DAML ,DAML-S
[68]	OWL,OWL-S
[69]	DAML ,DAML-S
[70]	OWL,OWL-S
[71]	OWL,OWL-S

**Tabla 0.2 Lista de ontologías**



### III. Lista de Algoritmos

Artículo	Técnica	campo
[18]	Algoritmo EHTN-DL que es un algoritmo híbrido de HTN-DL y WSMO	Planificación de Inteligencia Artificial Razonamiento sobre ontología
[19]	Teorema de la Lógica Lineal extendido con la técnica de Deducción Parcial	Planificación de Inteligencia Artificial
[20]	Algoritmo heurístico AO* y gráficos And/Or	Modelo de Representación/ Búsqueda
[21]	Uso de la ontología y su razonamiento semántico	Razonamiento sobre ontología
[22]	Uso de la ontología , coincidencia de palabras clave en redes P2P y el razonamiento semántico	Razonamiento sobre ontología
[23]	Teorema de lógica lineal (LL) y razonamiento semántico	Razonamiento Lógico Razonamiento sobre ontología
[24]	Probador de la lógica lineal (LL) y razonamiento semántico	Razonamiento sobre ontología Razonamiento Lógico
[25]	Ontología específica de dominio que contiene información funcional y no funcional y razonamiento semántico	Razonamiento sobre ontología
[26]	Event calculus	Razonamiento Lógico
[27]	Algoritmo HTN (Red jerárquica de tareas)	Planeamiento
[28]	La técnica de la terminología y La técnica extrínseca que son alineaciones de OWL-S	Razonamiento sobre ontología
[29]	Algoritmo de búsqueda informada basada en una función heurística	Planificación de Inteligencia Artificial Búsqueda
[30]	El algoritmo (IDDFS)	Búsqueda
[31]	Algoritmo backward or forward chaining	Razonamiento sobre ontología
[32]	Se utiliza el modelo de proceso OWL-S y el grado de coincidencia entre la descripción de la solicitud y el servicio.	Razonamiento sobre ontología
[33]	Algoritmo RPG	Modelo de Representación

[34]	Algoritmo forward-chaining	Razonamiento sobre ontología
[35]	Algoritmo de dos fases que realiza la composición del servicio web basado en entradas, salidas, precondiciones y efectos descritos utilizando OWL-S	Razonamiento sobre ontología
[36]	Gráfico de planificación RPG	Modelo de Representación
[37]	Coincidencia semántica. Matriz xCLM	Razonamiento sobre ontología Modelo de Representación
[38]	Lenguaje PDDL	Planificación de Inteligencia Artificial
[39]	Planificador de red de tarea jerárquica.	Planificación de Inteligencia Artificial
[40]	La técnica de terminología y la técnica extrínseca que son alineamientos de la ontología OWL-S	Razonamiento sobre ontología
[41]	Coincidencia en las características IOPE (entrada, salida, precondición y efecto) entre los servicios web semánticos. Diagramas de actividad UML	Razonamiento sobre ontología Flujo de Trabajo
[42]	Se planea utilizar métodos de planeamiento de la inteligencia artificial para la composición de artefactos Utiliza modelamiento de procesos	Planificación de Inteligencia Artificial Flujo de Trabajo
[43]	Se selecciona los servicios apropiados basados en funcional propiedades, propiedades funcionales y otras restricciones de usuario para generar composiciones ejecutables. Utiliza un planeador HTN para generar flujos de trabajo	Razonamiento sobre ontología Planeación de inteligencia artificial
[44]	Se basa en información semántica. La descripción de servicio es convertida en términos de lógica de primer orden	Razonamiento sobre ontología
[45]	Planificación utilizando PDDL Utiliza inferencia semántica	Planificación de Inteligencia Artificial Razonamiento sobre ontología
[46]	Se genera un flujo de trabajo personalizado de acuerdo con la ontología y al objetivo del usuario	Razonamiento sobre ontología

[47]	Coincidencia semántica OWL-S para encontrar posibles composiciones	Razonamiento sobre ontología
[48]	Técnica de razonamiento semántico	Razonamiento sobre ontología
[49]	Se busca composiciones de servicios web semánticos basado en repositorios de composiciones pasadas y un algoritmo que utiliza mecanismo de inferencia de la ontología de servicio	Razonamiento sobre ontología
[50]	Coincidencia semántica y uso de diagramas UML	Razonamiento sobre ontología Flujo de Trabajo
[51]	Metodología para descubrir composiciones de Servicios web que semántico tienen en cuenta tanto la información semántica como la información conductual publicado en las descripciones del servicio OWL-S.	Razonamiento sobre ontología
[52]	Información Semántica Algoritmo HTN	Razonamiento sobre ontología Planificación de Inteligencia Artificial
[53]	Uso de mecanismos de la Web Semántica junto con técnicas MDA para facilitar tanto Composición y reutilización del Servicio web semántico (SWS)	Razonamiento sobre ontología Flujo de Trabajo
[54]	Composición basada en la combinación entre el razonamiento basado en casos (CBR) y la información semántica	Inferencia Ontológica
[55]	Algoritmo de planeamiento global (HTN) Diagrama de estados UML Algoritmo de optimización	Planificación de Inteligencia Artificial Flujo de Trabajo
[56]	Composición basado en las anotaciones semánticas de los servicios	Razonamiento sobre ontología
[57]	Algoritmos basados en los lenguajes de lógica de descripción (DL).	Razonamiento sobre ontología
[58]	Se construye un planes de composición basados en algoritmos IA	Planificación de Inteligencia Artificial
[59]	Algoritmo BFS(breadthfirst search)	Planificación de Inteligencia Artificial
[60]	Se utiliza un algoritmo genético para evaluar y seleccionar servicios concretos para cada servicio abstracto tomando en cuenta restricciones QoS(Calidad de Servicio)	Búsqueda
[61]	Se transforma servicios semánticos en PDDL basado en técnicas de	Planificación de Inteligencia Artificial

	planeamiento	
[62]	Abductive Event Calculus para lograr un plan de composición	Planificación de Inteligencia Artificial
[63]	Descripciones semánticas de los servicios	Razonamiento sobre ontología
[64]	Explotar la información del servicio web (IOPE) , combinado con técnicas de lenguaje natural (NLP) que expresan la solicitud del usuario	Razonamiento sobre ontología
[65]	Se utiliza Xplan que es un planificador de búsqueda híbrido heurístico basado en FF-planner	Búsqueda
[66]	Teorema de lógica lineal y uso de la información semántica	Técnica de Inteligencia artificial Razonamiento sobre ontología
[67]	Teorema de lógica lineal Información semántica	Técnica de Inteligencia artificial Razonamiento sobre ontología
[68]	Enfoque automatizado de composición de servicios, basado en algoritmos que infieren dependencias entre servicios para construir composiciones , utilizando un estructura intermedia de enlaces semánticos	Razonamiento sobre ontología
[69]	Utiliza información semántica	Razonamiento sobre ontología
[70]	Criterio de composición basado en reglas de negocio y en lógica , se utiliza un mecanismo de razonamiento denominado backward chaining	Razonamiento sobre ontología
[71]	Se verifica la similitud de la capacidad , compatibilidad sintáctica y semántica de las entradas y salidas además de la inter-operatividad entre los tipos de servicios y se toma en cuenta las restricciones	Razonamiento sobre ontología Flujos de Trabajo

**Tabla 0.3 Lista de técnicas y/ algoritmos**

# Bibliografía

- [1] J. H. a. O. L. T. Berners-Lee, «The Semantic Web,» *Scientific Am.*, p. 28–37, May 2001.
- [2] A. S. J. A. M. K. V. R. A. P. R. Kaarthik Sivashanmugam, «Metadata and Semantics for Web Services and Processes,» de *Datenbanken und Informationssysteme*, 2003, pp. 245-272.
- [3] N. a. O. D. a. S. S. Guarino, «Handbook on Ontologies,» *What Is an Ontology?*, pp. 1-17, 2009.
- [4] V. B. A. Gomez, «Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods,» in *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods KRR5*, 1999.
- [5] T. Slimani, «A Study Investigating Typical Concepts and Guidelines for Ontology Building,» de *Journal of Emerging Trends in Computing and Information Sciences. Vol. 5, No. 12 December 2014, ISSN 2079-8407, pp.886-893*, 2011.
- [6] N. Casella, *Legal Ontology Engineering*, London: Springer , 2011.
- [7] J. V. V. L. I. O. V. L. S. Torma, «Semantic Web Service – A Survey,» Helsinki University of Technology Department of Computer Science and Engineering, Helsinki, 2008.
- [8] S. NAKAJIMA, «Verification of Web Service Flows with Model-Checking Techniques,» de *Proceedings of the First International Symposium on Cyber Worlds (CW02)*, 2002.
- [9] G. D. E. M. a. V. V. M. Cambronero, «A comparative study between WSCI, WS-CDL, and OWL-S,» de *IEEE International Conference on e-Business Engineering*, 2009.
- [10] J. D. ., E. M. T. P. ., F. H. L. Cabral, «Approaches to Semantic Web Services: An Overview and Comparisons,» 2004.
- [11] M. Klein, «XML,RFD y Relatives,» de *IEEE INTELLIGENT SYSTEMS*, 2001.
- [12] T. Schreiter, «An Introduction into Semantic Web Services,» University of Potsdam.
- [13] M. Klusch, «Semantic Web Service Description».
- [14] A. V. Sowmya Kamath S., «Semantic web services - discovery , selection and composition techniques».
- [15] A. E. ., S. F. ., M. K. ., A. M. T. ., E. R. W. ., D. Z. Witold Abramowicz, «Security Aspects in Semantic Web Services Filtering».
- [16] B. Kitchenham, «Guidelines for performing Systematic Literature Re- views in Software Engineering,» Version 2.3, Keele University and University of Durham, EBSE Technical Report, 2007.
- [17] M. N. D. Z. K. H. W. A. M. K. Maria Vargas-Vera, «Challenges on Semantic Web Service,» de *Social Dimensions of Semantic Technologies and Web Services*, pp. Chapter II ,pp 5-19.
- [18] S. G. H. T. a. W. M. N. W. K. a. S. Ibrahim, «Semantic Web Service Discovery and Composition



Based on AI Planning and Web Service Modeling Ontology,» de *IEEE Asia-Pacific Services Computing Conference*, 2008.

- [19] S. M. a. A. G. a. M. M. a. P. K. a. A. Haseeb, «Applying Semantic Web Service Composition for Action Planning in Multi-robot Systems,» de *Fourth International Conference on Internet and Web Applications and Services*, 2009.
- [20] H. t. J. a. Y. q. Guo, «Executable Workflow of Composition Architecture on Semantic Web Service,» de *International Conference on Management and Service Science*, 2010.
- [21] Z. Z. a. H. Y. a. L. R. a. W. W. a. L. Zengzhi, «A P2P-Based Semantic Web Services Composition Architecture,» de *IEEE International Conference on e-Business Engineering*, 2009.
- [22] L. J. a. X. B. a. Y. W. a. C. D. a. Z. P. a. W. Kehong, «SEWSIP: semantic based Web services integration in P2P,» de *IEEE International Workshop on Service-Oriented System Engineering (SOSE'05)*, 2005.
- [23] J. R. a. P. K. a. M. Matskin, «Logic-based Web services composition: from service description to process model,» de *Proceedings. IEEE International Conference on Web Services*, 2004.
- [24] A. Y. a. A. G. N. a. M. M. a. M. A. Seyyedi, «A new broker-based semantic Web service discovery framework for selecting and ranking suggested Web services,» de *IEEE 6th International Conference on Intelligent Computer Communication and Processing*, 2010.
- [25] L. D. a. J. W. a. Z. Hu, «ISCF: A Semantic Web Service Composition Framework Based on OAA,» de *The 3rd International Conference on Grid and Pervasive Computing - Workshops*, 2008.
- [26] L. C. a. X. Yang, «Applying AI Planning to Semantic Web Services for Workflow Generation,» de *First International Conference on Semantics, Knowledge and Grid*, 2005 .
- [27] Y. X. a. X. Z. a. X. Huang, «Automated Semantic Web Service Composition Based on Enhanced HTN,» de *Fifth IEEE International Symposium on Service Oriented System Engineering*, 2010.
- [28] A. B. a. K. B. a. A. Balla, «DA5DCSWS: A Distributed Architecture for semantic Web services Discovery and Composition,» de *8th International Conference for Internet Technology and Secured Transactions*, 2013.
- [29] T. W. a. S. B. a. M. K. a. K. Geihs, «Semantic Web Service Composition for Service-Oriented Architectures,» de *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, 2008.
- [30] S. B. a. T. W. a. K. Geihs, «Making a Fast Semantic Service Composition System Faster,» de *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services*, 2007.
- [31] Z. Y. a. R. G. a. C. M. a. J. B. Z. a. Z. S. a. L. Z. a. H. M. Lee, «Automating integration of manufacturing systems and services: a semantic Web services approach,» de *31st Annual Conference of IEEE Industrial Electronics Society*, 2005.
- [32] J. K. a. J. L. a. B. Lee, «Runtime Service Discovery and Reconfiguration Using OWL-S Based Semantic Web Service,» de *7th IEEE International Conference on Computer and Information Technology* , 2007.

- [33] J. A. G. L. a. D. A. O. Carranza, «Reactive Planning Model for Web Service Composition under Time Restrictions,» de *Third International Conference on Digital Society*, 2009.
- [34] G. C. a. J. B. Z. a. C. P. L. a. Z. Y. a. W. R. a. L. Zhuang, «Collaborative Virtual Enterprise Integration via Semantic Web Service Composition,» de *2nd IEEE Conference on Industrial Electronics and Applications*, 2007.
- [35] U. B. a. T. Mande, «Automated web service composition using semantic descriptions,» de *IEEE Asia-Pacific Services Computing Conference*, 2009 .
- [36] J. A. G. L. a. D. A. Ovalle, «Web Services Planning Agent in Dynamic Environments with Incomplete Information and Time Restrictions,» de *11th IEEE International Conference on Computational Science and Engineering - Workshops*, 2008.
- [37] F. L. a. A. Leger, «Semantic Web Service Composition through a Matchmaking of Domain,» de *European Conference on Web Services (ECOWS'06)*, 2006 .
- [38] J. a. L. A. a. S. M. a. L. J. Puttonen, «Cloud computing as a facilitator for web service composition in factory automation,» de *Journal of Intelligent Manufacturing*, 2016.
- [39] J. J. P. A. D. E. B. Juan Carlos Zuñiga, «A loosely coupled architecture for automatic composition of web services applications,» de *International Journal of Metadata, Semantics and Ontologies*, 2014.
- [40] A. a. B. K. a. B. A. Boukhadra, «Automatic composition of semantic Web services-based alignment of OWL-S,» de *CEUR Workshop Proceedings*, 2012.
- [41] C. Panl, «Semantic model-driven development of web service architectures,» de *International Journal of Web Engineering and Technology*, 2008.
- [42] I. a. M. I. a. D. C. Weber, «A conceptual framework for composition in business process management,» de *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007.
- [43] U. a. P. S. Arul, «A unified algorithm to automatic semantic composition using multilevel workflow orchestration,» de *Cluster Computing*, 2018.
- [44] S. a. B. A. a. G. G. a. B. M. Bansal, «Generalized semantic Web service composition,» de *Service Oriented Computing and Applications*, 2016.
- [45] T. a. G. H. a. H. H. a. N. M. Fissaa, «How can semantics and context awareness enhance the composition of context-aware services?,» de *17th International Conference on Enterprise Information Systems, Proceedings*, 2015 .
- [46] C.-H. a. C. J.-Y. Liu, «Using ontology-based BDI agent to dynamically customize workflow and bind semantic web service,» de *Journal of Software*, 2012.
- [47] M. a. B. M. Gharzouli, «A distributed P2P-based architecture for semantic web services discovery and composition,» de *NOTERE'10 - 10th Annual International Conference on New Technologies of Distributed Systems*, 2010.
- [48] T. a. B. P. a. M. R. a. F. R. a. J. A. Darr, «Semantic services for intelligence preparation of the battlefield (IPB) composition,» de *International Symposium on Collaborative Technologies and Systems*, 2010 .

- [49] L. a. X. Z. a. Y. Q. Han, «An approach to web service composition based on service-ontology,» de *6th International Conference on Fuzzy Systems and Knowledge Discovery*, 2009.
- [50] Z. a. C. Y. a. L. R. a. L. Z. Zhu, «Study of mda based semantic web service composition,» de *Information Technology Journal*, 2009.
- [51] F. a. B. A. a. C. S. Benigni, «Discovering service compositions that feature a desired behaviour,» de *Memorias de la 11th Conferencia Iberoamericana de Software Engineering*, 2008.
- [52] A. a. E.-G. O. Deokar, «A semantic web services-based architecture for model management systems,» de *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2008.
- [53] D. M. L. B. G. Carlos Granell, «An ontology-driven architecture for re-using semantic web services,» de *Lecture Notes in Computer Science*, 2007.
- [54] J. M. G.-B. B. D.-A. a. P. A. G.-C. J. A. Recio-García, «The WINGS of jCOLIBRI: A CBR architecture for semantic web services,» de *3rd International Conference on Next Generation Web Services Practices*, 2007.
- [55] L. a. C. L. a. L. F. a. S. Z. Qiu, «Context optimization of AI planning for semantic Web services composition,» de *Service Oriented Computing and Applications*, 2007.
- [56] E. a. P. B. a. H. J. Sirin, «Composition-driven filtering and selection of semantic Web services,» de *AAAI Spring Symposium - Technical Report*, 2004.
- [57] C. a. L. S. a. T. R. a. W. B. a. O. N. Baluz, «Composition of web services in the ICS architecture,» de *Proceedings of the Sixth International Conference on Enterprise Information Systems*, 2004 .
- [58] A. a. B. M. a. C. D. a. B. B. Adadi, «A semantic web service composition for E-Government services,» de *Journal of Theoretical and Applied Information Technology*, 2015.
- [59] J. a. L. A. a. C. S. M. a. M. L. J. Puttonen, «Planning-based semantic web service composition in factory automation,» de *Advanced Engineering Informatics*, 2015.
- [60] M. a. L. H. a. C. Q. Lu, «A method of semantic web service automatic composition based on genetic algorithm,» de *Advances in Intelligent and Soft Computing*, 2012.
- [61] I. a. K. S. a. F. F. Zinnikus, «An integration of semantically enabled service oriented architectures and agent platforms,» de *Lecture Notes in Business Information Processing*, 2012.
- [62] D. a. S. S. Paulraj, «Composition of composite semantic web services using abductive event calculus,» de *Communications in Computer and Information Science*, 2011.
- [63] P. a. B. B. Larvet, «Discovery and auto-composition of semantic web services,» de *1st International Conference on Software and Data Technologies, Proceedings*, 2006.
- [64] A. a. C. F. a. V. G. a. M. R. Bosca, «On-the-fly construction of Web Services compositions from Natural Language requests,» de *Journal of Software*, 2006.
- [65] C. C. A. F. N. F. H. H. O. K. A. K. M. K. H. L. A. L. S. O. H. S. M. S. Federico Bergenti, «Context-aware service coordination for mobile e-health applications,» de *European Conference on eHealth 2006*, 2006.

- [66] P. K. M. Jinghai Rao, «Composition of semantic web services using linear logic theorem proving,» de *Journal Information Systems*, 2006.
- [67] J. a. S. X. Rao, «Toward the composition of semantic web services,» de *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2004.
- [68] F. L. M. Rik Eshuis, «Flexible Construction of Executable Service Compositions from Reusable Semantic Knowledge,» de *Journal ACM Transactions on the Web (TWEB)*, 2016.
- [69] B. a. B. A. a. E. A. K. Medjahed, «Composing Web Services on the Semantic Web,» de *The International Journal on Very Large Data Bases*, 2003.
- [70] Y. a. J. M. Gong, «Creating Dynamic Business Processes Using Semantic Web Services and Business Rules,» de *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, 2011.
- [71] E. a. K. K. a. S. P. Karakoc, «A Workflow-Based Web Service Composition System,» de *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2006.

