

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**ESCUELA DE POSGRADO**



**Título**

**Mapeo sistemático sobre las arquitecturas de software en el  
desarrollo ágil**

**TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER EN  
INFORMÁTICA CON MENCIÓN EN INGENIERÍA DE SOFTWARE**

**AUTOR**

Naldo Reupo-Musayón Gastulo

**ASESOR**

Mag. Dennis Stephen Cohn Muroy

Noviembre, 2020

## DEDICATORIA

Al regalo más grande que me entregó la vida, mi hija Rubí y a mi abuela Violeta, aunque no pueda verla, vive en mis recuerdos y sé que me cuida.



## **AGRADECIMIENTOS**

A mis padres Blanca y Naldo por su apoyo incondicional y constante.

A mi hija Rubí por darle sentido a mi vida y ser la causante de mis ganas de salir adelante.

A mis hermanas Isabel y Pierina, por creer en mí y siempre alentarme.

A mi asesor Dennis Cohn, por sus acertados y útiles consejos en la dirección de esta investigación.



## RESUMEN

(ANTECEDENTES) El uso de *frameworks* y metodologías ágiles en el desarrollo de *software* es cada vez mayor, priorizando la entrega de valor al cliente, en este contexto las actividades de arquitectura de *software* son omitidas al no entregar un valor tangible, existiendo un aparente conflicto de perspectivas y no se tiene definido cuanto esfuerzo se debe invertir en el desarrollo de una arquitectura en proyectos ágiles.

(OBJETIVOS) El objetivo de este trabajo es consolidar las distintas investigaciones respecto al uso de arquitecturas de *software* en el desarrollo ágil, identificar patrones arquitectónicos, factores, beneficios, desafíos, y lecciones aprendidas con respecto a la combinación.

(MÉTODOS) Para este estudio se realizó un mapeo sistemático de la literatura en bases de datos digitales relevantes.

(RESULTADOS) Se seleccionaron 61 artículos publicados desde el año 2015 hasta el año 2020, el 54% fueron de aplicación industrial principalmente en el sector salud, aeroespacial y automotriz, se pudo identificar que en el año 2016 se publicaron el mayor número de artículos referente al tema de investigación, donde la conferencia es el tipo de publicación más utilizado y el evento *IEEE International Conference* es el mayor canal de distribución. Adicionalmente, se identificó que el estilo arquitectónico más empleado es SOA, la práctica ágil más referenciada es Scrum, el uso combinado del *framework* Scrum y el estilo SOA es el más usado, emplear el estilo SOA en el sector salud es el más citado en las publicaciones, la flexibilidad que brinda tener una arquitectura sólida es la mayor ventaja referenciada asimismo los conflictos de enfoques entre la agilidad y las actividades de arquitectura es identificado como el mayor inconveniente que se afronta, y la comunicación es el factor que más influye en la adopción de arquitecturas de *software* en el desarrollo ágil.

### Palabras clave

Arquitectura de *software*, Arquitectura de aplicaciones, Arquitectura ágil, Agilidad, desarrollo de *software*, ingeniería de *software*, diseño de *software*

## **ABSTRACT**

*(BACKGROUND) The use of agile frameworks and methodologies in software development is increasing, prioritizing the delivery of value to the client, in this context, software architecture activities are omitted by not delivering tangible value, with an apparent conflict of perspectives and it is not defined how much effort should be invested in the development of an architecture in agile projects.*

*(OBJECTIVES) The objective of this work is to consolidate the different investigations regarding the use of software architectures in agile development, to identify architectural patterns, factors, benefits, challenges, and lessons learned regarding the combination.*

*(METHODS) For this study, a systematic mapping of the literature in relevant digital databases was carried out.*

*(RESULTS) 61 articles published from 2015 to 2020 were selected, 54% were of industrial application mainly in the health, aerospace, and automotive sectors, it was possible to identify that in 2016 the largest number of articles were published on the subject of research, where the conference is the most used type of publication and the IEEE International Conference event is the largest distribution channel. Additionally, it was identified that the most used architectural style is SOA, the most referenced agile practice is Scrum, the combined use of Scrum framework and the SOA style is the most used, using the SOA style in the health sector is the most cited in publications, the flexibility provided by having a solid architecture is the greatest advantage referenced also the conflicts of approaches between agility and architectural activities is identified as the greatest inconvenience faced, and communication is the factor that most influences the adoption of software architectures in agile development.*

## **Keywords**

*Software Architecture, Application Architecture, Agile Architecture, Agility, Software Development, Software Engineering, Software Design*

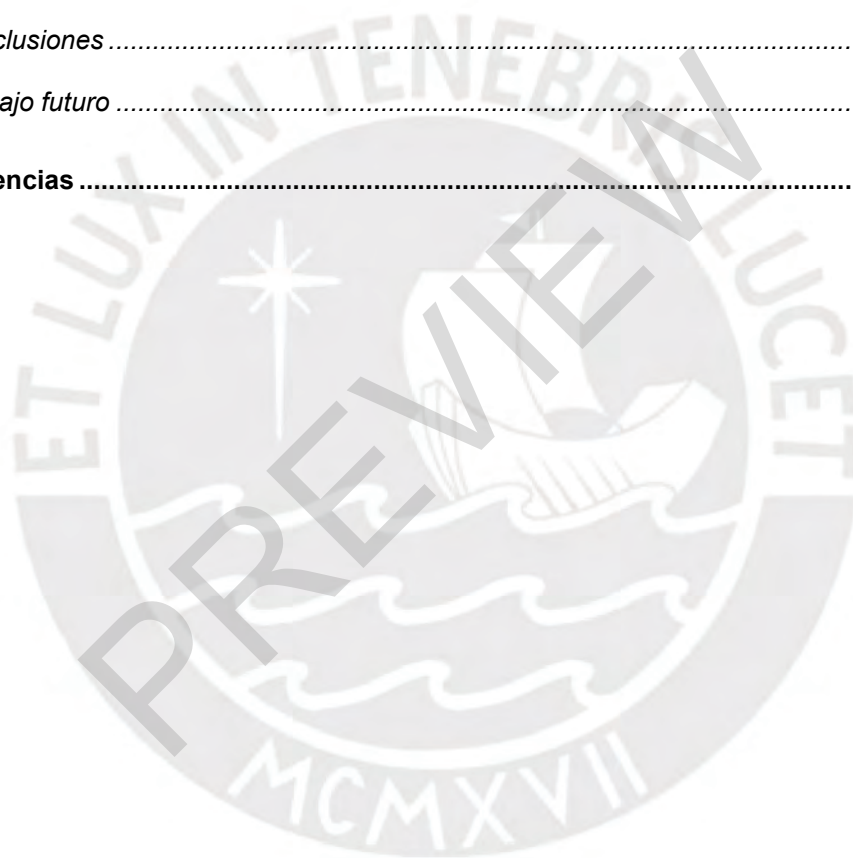
## CONTENIDO

<b>Dedicatoria .....</b>	<b>ii</b>
<b>Agradecimientos .....</b>	<b>iii</b>
<b>Resumen .....</b>	<b>iv</b>
<b>Abstract .....</b>	<b>v</b>
<b>Contenido.....</b>	<b>vi</b>
<b>Índice de tabla .....</b>	<b>ix</b>
<b>Índice de figuras.....</b>	<b>x</b>
<b>1. INTRODUCCIÓN .....</b>	<b>1</b>
1.1. Contexto del problema .....	1
1.2. Justificación .....	2
1.3. Objetivos del trabajo.....	3
<b>2. Marco de Teórico.....</b>	<b>6</b>
2.1. La arquitectura de software .....	6
2.1.1. Arquitectura Orientada a Servicios.....	6
2.1.2. N-Capas.....	6
2.1.3. MVC .....	7
2.1.4. Microservicios .....	8
2.1.5. Arquitectura orientada a Aspectos .....	9
2.1.6. Basado en componentes.....	9
2.1.7. DCI.....	10
2.2. Metodologías y frameworks ágiles .....	10
2.2.1. SAFe .....	11
2.2.2. Scrum .....	11
2.2.3. Extreme Programming.....	11
2.2.4. Lean software development .....	11
<b>3. Estado del Arte .....</b>	<b>12</b>
<b>4. Planificación .....</b>	<b>13</b>
4.1. Metodología.....	13



4.2.	<i>Preguntas de bibliometría.....</i>	14
4.3.	<i>Preguntas de investigación .....</i>	14
4.4.	<i>Estrategia de búsqueda.....</i>	16
4.5.	<i>Cadenas de búsqueda .....</i>	17
4.6.	<i>Criterios de selección de estudios.....</i>	18
<b>5.</b>	<b>Conducción.....</b>	<b>20</b>
5.1.	<i>Selección de estudios.....</i>	20
5.2.	<i>Evaluación de calidad.....</i>	20
5.3.	<i>Extracción de los datos .....</i>	29
<b>6.</b>	<b>análisis de Resultados.....</b>	<b>37</b>
6.1.	<i>Preguntas Bibliométricas.....</i>	37
	PB-1 ¿Qué proporción de las investigaciones son académicas y de aplicación industrial, en que sectores industriales se investiga y/o aplica? .....	37
	PB-2 ¿Cómo ha ido evolucionando el número de publicaciones relacionados al tema de investigación? .....	38
	PB-3 ¿Qué tipos de estudios se realizan con mayor frecuencia con relación al tema trazado? ..	39
	PB-4 ¿Qué canales de publicación (revistas y/o eventos) son los principales objetivos para las investigaciones relacionados al tema de investigación? .....	40
6.2.	<i>Preguntas de investigación .....</i>	43
	PI-1 ¿Cuáles son las arquitecturas de <i>software</i> que pueden utilizarse en el desarrollo ágil? .....	43
	PI-2 ¿Cuáles son los <i>frameworks</i> /metodologías ágiles? .....	45
	PI-3 ¿En qué ámbitos de la industria tienden a utilizar en combinación arquitecturas de <i>software</i> y <i>frameworks</i> ágiles considerando proyectos ágiles? .....	46
	PI-4 ¿Cuáles son las ventajas y beneficios en la adopción de una arquitectura de <i>software</i> en el desarrollo ágil? .....	48
	PI-5 ¿Cuáles son las desventajas e inconvenientes en la adopción de una arquitectura de <i>software</i> en el desarrollo ágil? .....	50
	PI-6 ¿Cuáles son los factores que influyen adopción de una arquitectura de <i>software</i> en el desarrollo ágil? .....	52
	PI-7 ¿Cuáles son las relaciones entre arquitecturas de <i>software</i> y <i>frameworks</i> /metodologías ágiles? .....	53
	PI-8 ¿Cuáles son las relaciones entre las arquitecturas de <i>software</i> y las industrias? .....	55
	PI-9 ¿Cómo han evolucionado las principales arquitecturas de <i>software</i> en el desarrollo ágil a través de los años? .....	57

PI-10 ¿Cómo ha evolucionado los principales <i>frameworks</i> /metodologías usados a través de los años? .....	58
<b>7. Amenazas a la validez.....</b>	<b>60</b>
7.1. Validez interna .....	60
7.2. Validez de las conclusiones .....	60
7.3. Validez del constructo .....	60
7.4. Validez externa .....	60
<b>8. Conclusiones y trabajo futuro .....</b>	<b>61</b>
8.1. Conclusiones .....	61
8.2. Trabajo futuro .....	61
<b>9. Referencias .....</b>	<b>63</b>





## ÍNDICE DE TABLA

Tabla 1 PICO .....	16
Tabla 2 Cadenas de búsqueda .....	18
Tabla 3 Criterios de Inclusión.....	19
Tabla 4 Criterios de Exclusión .....	19
Tabla 5 Criterios de selección.....	20
Tabla 6 Criterios de calidad .....	21
Tabla 7 Aplicación de criterios de calidad .....	28
Tabla 8 Selección de artículos .....	29
Tabla 9 Estudios seleccionados .....	36
Tabla 10 Artículos clasificados por sector.....	37
Tabla 11 Artículos clasificados por año .....	38
Tabla 12 Artículos clasificados por tipo de estudio .....	39
Tabla 13 Artículos clasificados por revistas y/o evento .....	43
Tabla 14 Artículos clasificados por patrón de arquitectura.....	44
Tabla 15 Artículos clasificados por <i>frameworks</i> /metodologías .....	46
Tabla 16 Artículos clasificados por industria .....	47
Tabla 17 Artículos clasificados por ventajas y beneficios.....	49
Tabla 18 Artículos clasificados por desventajas e inconvenientes .....	51
Tabla 19 Artículos clasificados por factores de adopción.....	53
Tabla 20 Artículos clasificados por la combinación de arquitectura de <i>software</i> y <i>frameworks</i> /metodología ágil .....	54
Tabla 21 Artículos clasificados por la combinación de arquitectura de <i>software</i> e industrias .....	57
Tabla 22 Evolución de las principales arquitecturas de <i>software</i> en el desarrollo ágil a través de los años .....	58
Tabla 23 Evolución de los principales <i>frameworks</i> /metodologías usados a través de los años .....	59

## ÍNDICE DE FIGURAS

Ilustración 1 Árbol de problemas .....	2
Ilustración 2 Árbol de soluciones .....	4
Ilustración 3 Patrón N-Capas.....	7
Ilustración 4 Patrón MVC .....	8
Ilustración 5 Microservicios .....	9
Ilustración 6 Diagrama de flujo proceso de mapeo sistemático. ....	13
Ilustración 7 Selección de artículos.....	29
Ilustración 8 Artículos clasificados por sector .....	37
Ilustración 9 Artículos clasificados por año .....	39
Ilustración 10 Artículos clasificados por tipo de estudio .....	40
Ilustración 11 Artículos clasificados por patrón de arquitectura.....	45
Ilustración 12 Artículos clasificados por <i>frameworks</i> /metodologías .....	46
Ilustración 13 Artículos clasificados por industria.....	48
Ilustración 14 Artículos clasificados por ventajas y beneficios .....	50
Ilustración 15 Artículos clasificados por desventajas e inconvenientes .....	51
Ilustración 16 Artículos clasificados por factores de adopción.....	53
Ilustración 17 Artículos clasificados por la combinación de arquitectura de <i>software</i> y <i>frameworks</i> /metodología ágil .....	55
Ilustración 18 Artículos clasificados por la combinación de arquitectura de <i>software</i> e industrias.....	57
Ilustración 19 Evolución de las arquitecturas de <i>software</i> en el desarrollo ágil a través de los años .....	58
Ilustración 20 Evolución de los principales <i>frameworks</i> /metodologías usados a través de los años .....	59

# 1. INTRODUCCIÓN

## 1.1. Contexto del problema

El uso de *frameworks* y metodologías ágiles en la construcción de *software* en diversas industrias ha comenzado a crecer sostenidamente en los últimos años. Por ejemplo, *CollabNet VersionOne* [1] en su más reciente encuesta de más de 1,300 empresas, el 97% indicó que usa *frameworks* ágiles en sus desarrollos; sin embargo, muchas organizaciones luchan con la toma de decisiones de arquitectura [2] dado que las actividades de arquitectura de *software* no se analizan en la mayoría de los métodos de desarrollo de *software* ágiles[3] y el tiempo invertido en arquitectura es cada vez más crucial [4]. Si el equipo pasa demasiado tiempo diseñando la arquitectura la entrega de valor al cliente se retrasa [5]. El diseño de una arquitectura es demasiado trabajo sin un beneficio claro según los agilistas [6]; generando una deuda técnica en los proyectos - que es uno de las principales problemas en los proyectos ágiles [7] - siendo percibida la elección de una arquitectura inadecuada como el principal factor en los proyectos de *software* [8].

No se tiene definido cuanto diseño es suficiente en el desarrollo ágil [9], esto genera un conflicto entre definir una arquitectura inicial completa o construir la arquitectura gradualmente durante desarrollo del producto [10]: adaptación versus anticipación [11] .

El problema central encontrado es que el desarrollo ágil es altamente vulnerable a deuda técnica por deficiencias de arquitectura [12].

Las causas que explican el problema planteado en la investigación son:

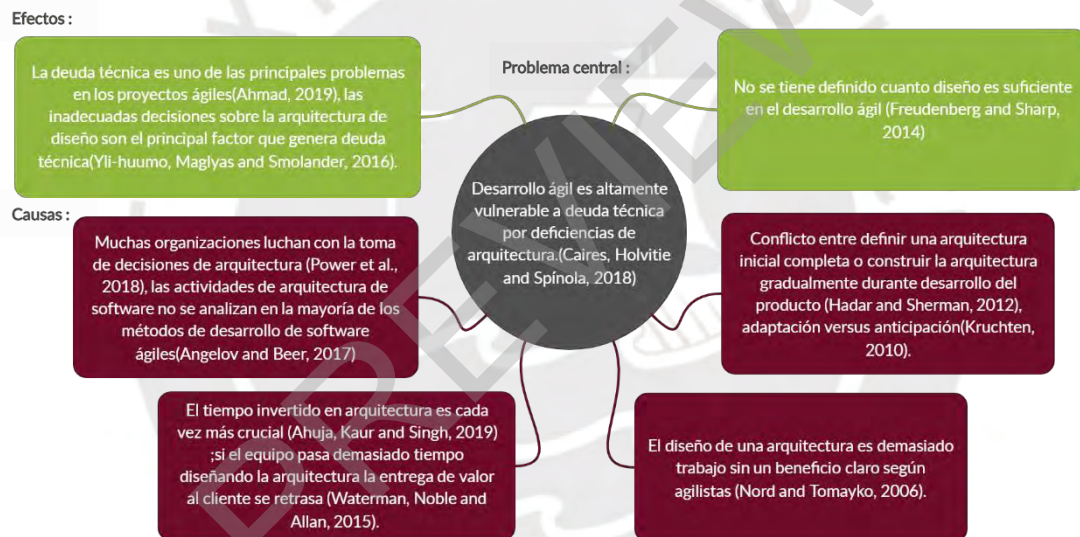
- Muchas organizaciones luchan con la toma de decisiones de arquitectura [13], las actividades de arquitectura de *software* no se analizan en la mayoría de los métodos de desarrollo de *software* ágiles[3].
- El tiempo invertido en arquitectura es cada vez más crucial [4] ;si el equipo pasa demasiado tiempo diseñando la arquitectura la entrega de valor al cliente se retrasa [5].
- El diseño de una arquitectura es demasiado trabajo sin un beneficio claro según los agilistas [6].

- Conflicto entre definir una arquitectura inicial completa o construir la arquitectura gradualmente durante desarrollo del producto [10]: adaptación versus anticipación[14].

Los efectos que derivan del problema central son:

- La deuda técnica es uno de los principales problemas en los proyectos ágiles[7], las inadecuadas decisiones sobre la arquitectura de diseño son el principal factor que genera deuda técnica[15].
- No se tiene definido cuanto diseño es suficiente en el desarrollo ágil [9].

A partir del problema planteado, las causas y efectos descritos anteriormente se propone el árbol de problemas mostrado en la Ilustración 1.



**Ilustración 1 Árbol de problemas**

## 1.2. Justificación

El desarrollo ágil enfrenta el dilema de determinar cuánto esfuerzo invierte en la arquitectura de *software* [16] , por eso surge la necesidad de un estudio que permita explorar y analizar los artículos en relación con el uso de arquitecturas en el desarrollo ágil para identificar los patrones y estilos arquitectónicos de *software* compatibles con prácticas ágiles, describir las relaciones, factores, beneficios e inconvenientes en el uso de arquitecturas y *frameworks* ágiles.

### 1.3. Objetivos del trabajo

El objetivo general es realizar un mapeo sistemático que consolide las distintas investigaciones respecto al uso de arquitecturas de *software* en el desarrollo ágil, identificar patrones arquitectónicos, factores, beneficios, desafíos, y lecciones aprendidas con respecto a la combinación.

Los objetivos específicos para esta investigación son:

- OE1: Identificar los patrones arquitectónicos compatibles con los *frameworks* y metodologías ágiles.
- OE2: Identificar *frameworks* y metodologías ágiles compatibles con arquitecturas de *software*.
- OE3: Identificar los ámbitos de las industrias que utilizan la combinación de arquitectura de *software* y *frameworks*/metodologías ágiles.
- OE4: Describir las ventajas y beneficios en la adopción de una arquitectura de *software* en el desarrollo ágil.
- OE5: Describir las desventajas e inconvenientes en la adopción de una arquitectura de *software* en el desarrollo ágil.
- OE6: Determinar los factores claves que influyen en la adopción de una arquitectura de *software* en el desarrollo ágil.
- OE7: Determinar las relaciones entre los patrones de arquitectura y los *frameworks* ágiles.

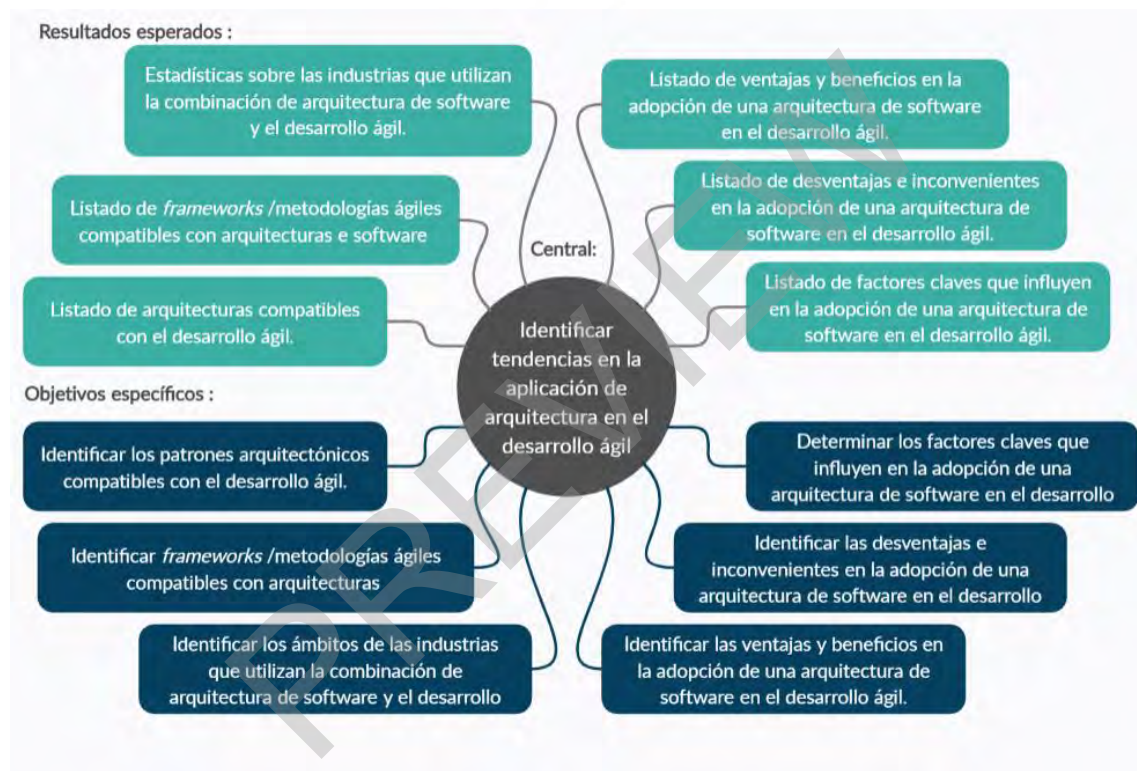
Los resultados que se esperan para esta investigación son:

- R1: Listado de arquitecturas compatibles con el desarrollo ágil (OE1).
- R2: Listado de *frameworks* y metodologías ágiles compatibles con arquitecturas de *software* (OE2).
- R3: Estadísticas sobre las industrias que utilizan la combinación de arquitectura de *software* y el desarrollo ágil (OE3).
- R4: Listado de ventajas y beneficios en la adopción de una arquitectura de *software* en el desarrollo ágil (OE4).



- R5: Listado de desventajas e inconvenientes en la adopción de una arquitectura de *software* en el desarrollo ágil (OE5).
- R6: Listado de factores claves que influyen en la adopción de una arquitectura de *software* en el desarrollo ágil (OE6).
- R7: Estadísticas sobre las relaciones entre los patrones de arquitectura y el desarrollo ágil (OE7).

A partir de los objetivos y resultados descritos anteriormente se elabora el árbol de soluciones que se muestra en la Ilustración 2.

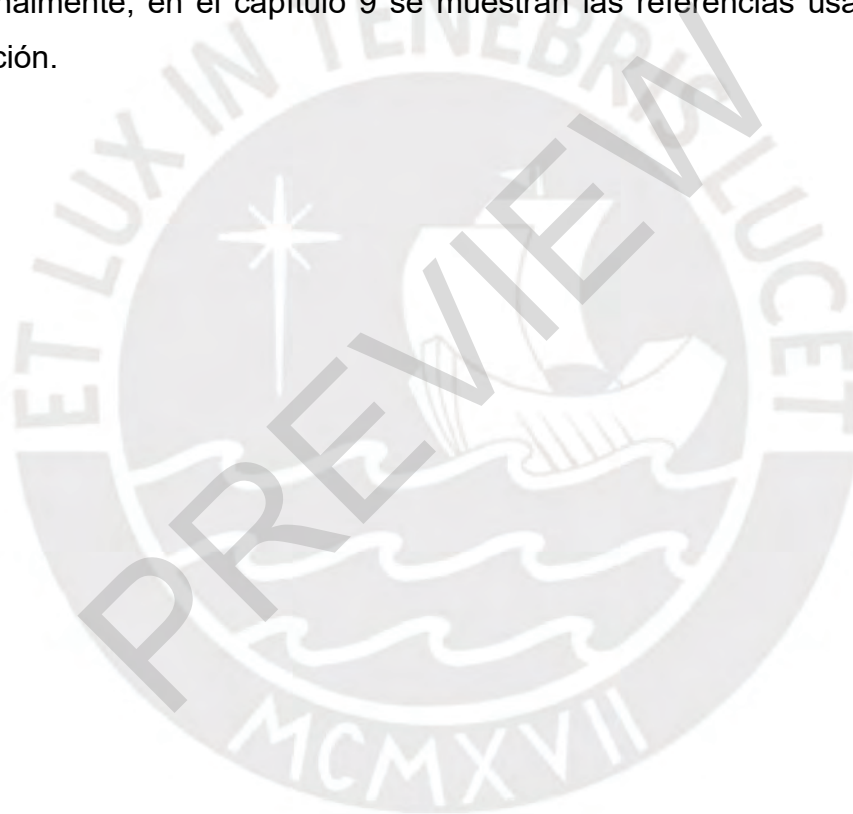


**Ilustración 2 Árbol de soluciones**

El actual estudio se encuentra constituido de la siguiente manera:

El capítulo 1 indica el contexto del problema, la justificación de la investigación y los objetivos del trabajo. En el capítulo 2 se presenta el marco teórico donde se describen las arquitecturas de *software* y *frameworks*/metodologías usadas en la investigación. En el capítulo 3 se muestra el estado del arte donde se analizan las 5 publicaciones relacionadas con el presente trabajo. En el capítulo 4 se

detalla la planificación del mapeo sistemático, se define la metodología, se plantean las preguntas de bibliometría e investigación, se define la estrategia de búsqueda y las cadenas de búsqueda, finalmente se plantean los criterios de inclusión y exclusión para la de selección de estudios. En el capítulo 5 se documenta la conducción del estudio, se presenta el detalle de la selección de estudios, se aplican los criterios de evaluación de calidad, finalmente se ejecuta la extracción de los datos. En el capítulo 6 se hace el análisis de resultados, se responden las preguntas bibliométricas y de investigación. En el capítulo 7 se analizan las amenazas a la validez interna, validez externa, validez del constructo y validez las conclusiones. En el capítulo 8 conclusiones y trabajo futuro. Finalmente, en el capítulo 9 se muestran las referencias usadas en la investigación.





## **2. MARCO DE TEÓRICO**

En esta sección se presenta la definición de arquitectura de *software* y los patrones utilizados; así como la definición de metodología ágil y las metodologías usadas en la presente investigación.

### **2.1. La arquitectura de *software***

La arquitectura de *software* es el conjunto de estructuras necesarias para que funcione el sistema, comprende los elementos del *software*, propiedades y las relaciones entre ellos [17], es el diseño de más alto nivel en un *software* y debe describir los diferentes aspectos del sistema.

Existen diversos paradigmas y diseños arquitectónicos. Adoptar una arquitectura dependerá de la necesidad y recursos con los que cuente en proyecto. Los patrones de arquitectura referenciados en la investigación son:

#### **2.1.1. Arquitectura Orientada a Servicios**

Es un estilo arquitectónico que modularizó el sistema de información en servicios. Con SOA, estos importantes programas se convierten en servicios comerciales. Con un único servicio comercial para una función determinada que se utiliza en todas partes de la organización. Cuando se necesita cambiar la política de negocios, se puede cambiar en un lugar y debido a que el mismo servicio se usa en todas partes, la consistencia se mantendrá en toda la organización. SOA permite a las empresas tomar decisiones comerciales respaldadas por la tecnología en lugar de tomar decisiones comerciales determinadas o limitadas por la tecnología[18].

La agilidad, adaptación e independencia de plataforma son las principales ventajas de esta arquitectura.

#### **2.1.2. N-Capas**

Separa los componentes de su solución en capas. Los componentes de cada capa deben ser cohesivo y tener aproximadamente el mismo nivel de abstracción [19].

Las capas son agrupaciones horizontales lógicas de componentes de software que forman la aplicación o el servicio. Nos ayudan a diferenciar entre los