



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

**Gestión de Recursos de Aprendizaje
para generar Servicios Web de Aprendizaje**

presentada por

Ing. Iván Humberto Fuentes Chab

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Directora de tesis

Dra. Olivia Graciela Fragoso Díaz

Cuernavaca, Morelos, México. Noviembre de 2019.



"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor., 25/octubre/2019
Oficio No. DCC/085/2019
Asunto: Aceptación de documento de tesis

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del Ing. Iván Humberto Fuentes Chab, con número de control M17CE089, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "Gestión de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dra. Olivia Graciela Fragoso Díaz
Doctora en Ciencias en Ciencias
de la Computación
7420199

REVISOR 1

M.C. Humberto Hernández García
Maestro en Ciencias con
Especialidad
en Sistemas Computacionales
7573641

REVISOR 2

Dr. Juan Carlos Rojas Pérez
Doctor en Ciencias en Ciencias de
la Computación
6099372

C.p. M.E. Guadalupe Garrido Rivera - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

JGGS/lmz



"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor.,

No. de Oficio:

Asunto:

31/octubre/2019

SAC/288/2019

**Autorización de
impresión de Tesis**

**ING. IVÁN HUMBERTO FUENTES CHAB
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "Gestión de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

*Excelencia en Educación Tecnológica
"Conocimiento y tecnología al servicio de México"*

**DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO**



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.E. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr



Dedicatoria

“Los hombres son lo que sus madres hicieron de ellos”
(Ralph Waldo Emerson, 1803 - 1882)

Para mi madre **Elsy Guadalupe** y mi padre **José Armando** por todo su apoyo incondicional durante la realización de este trabajo. A ambos por proporcionarme de su amor y educación a lo largo de mi vida. Con mucho amor y cariño, este logro también es de ustedes.

Para mis hermanos **Aldo** y **Daniela** por su apoyo y cariño. Gracias por sus consejos, palabras de aliento y todos los bellos momentos que he pasado con ustedes. Cuentan siempre conmigo.

Para mis abuelos(as) y tíos(as) que me proporcionaron amor y apoyo a lo largo de mi vida. A aquellos que hoy no están físicamente, pero siempre me acompañarán en mente y corazón. Y por supuesto, a mis queridos sobrinos que alegran la vida con sus bellas sonrisas.

Para mis mejores amigos **Violeta, Mildred, Damián, Pablo, Oswaldo, Axel, Xicotécatl, Luis** y **Gerardo** que me acompañaron durante este proceso de formación académica y profesional.

Para mi novia **Alondra** por su atención, paciencia y amor que me ha brindado.

Para ustedes con todo mi cariño

Agradecimientos

Al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por otorgarme el sustento económico para realizar este trabajo de investigación.

Al **Tecnológico Nacional de México (TecNM) / Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)**, a su personal docente y administrativo por brindarme la oportunidad de realizar mis estudios de maestría en esta destacada institución.

A mi directora de tesis, **Dra. Olivia Graciela Fragoso Díaz**, por su tiempo, apoyo, motivación, paciencia y conocimientos para la realización de este trabajo. Gracias por su confianza y por aportar en gran medida mi formación profesional.

A mi comité revisor conformado por el **Dr. Juan Carlos Rojas Pérez** y el **M.C. Humberto Hernández García**, por su tiempo y atención brindados en el desarrollo de este trabajo para engrandecer el contenido a través de sus valiosas aportaciones.

Al **Cuerpo Académico de Ingeniería de Software (CAIS)** por sus enseñanzas transmitidas en cada una de las clases y asesorías que me brindaron.

A la **Universidad de Medellín (UdeM)**, a su personal docente y administrativo por brindarme la oportunidad de realizar mi estancia de movilidad durante Julio del 2019 y a mi tutora, **Dra. María Clara Gómez Álvarez**, por compartir sus conocimientos y aportaciones a este trabajo.

A **mi familia** por todo su apoyo, confianza y amor brindado para seguir adelante y concluir con éxito esta etapa de mi vida. Especialmente a mi padre **José Armando Fuentes Sulú**, mi madre **Elsy Guadalupe Chab Valle**, mi hermano **Aldo Armando** y mi pequeña hermana **Daniela Guadalupe**.

A mi novia **Alondra Méndez** por todo el apoyo, motivación y cariño otorgado a pesar de la distancia.

A todos los amigos y compañeros que conocí en el CENIDET, en especial a los de mi generación por llevar esta etapa de formación conmigo, así como compartir gratos y alegres momentos: **Heidi Barrera, Violeta Bautista, Santiago Hernández, Cristian López, Ricardo López, Xicotécatl Nogueta y Juan Carlos Soto**.

A mis romíos con los que compartí hermosos y divertidos momentos: **Uriel Bahena, Oswaldo Barrera, Julio Bernabé, Mario Domínguez, Luis Montalvo, Pablo Padilla, Felipe Pérez, Marisol Ramírez, Axel Rojas, Gerardo Romero, Julio Torres, Mildred Xicotécatl**, y en especial a **Damián Rosado** quién me sugirió estudiar la maestría.

A **Chachi**, la cachorra que apareció después del temblor del 19 de septiembre del 2017 y llegó a ser parte de mi vida desde entonces.

A todos por ser parte de este importante logro, ¡gracias totales!

Resumen

El desarrollo de las aplicaciones web ha ido evolucionando con el paso de los años, de tal manera que cada vez resulten más sencillas de usar, pero también que aumenten sus funciones. Los servicios web surgen para brindar ciertas ventajas al usuario, evitando la dependencia entre módulos o componentes de las aplicaciones.

Las temáticas de las aplicaciones y los servicios web pueden ser de diversa índole, una de las áreas más demandadas en la actualidad es la educación, donde podemos encontrar mucho contenido educativo disponible en Internet, llamados también Recursos de Aprendizaje. En este contexto, se define como aprendizaje electrónico (e-learning) al proceso de aprendizaje-enseñanza a través del uso de las Tecnologías de Información y Comunicación (TICs). Una forma en la que se estructura el contenido educativo es mediante Objetos de Aprendizaje, los cuales están compuestos de cuatro elementos pedagógicos conocidos como: objetivo de aprendizaje, contenido informativo, actividades de aprendizaje y evaluación.

Con motivo de lo anterior descrito, se da origen a los Servicios Web de Aprendizaje (SWA) para contener alguno de estos tipos de objetos de aprendizaje en un servicio web para ser entregado al usuario [Delgado Fernández, 2012]. Estos SWA contienen un Recurso de Aprendizaje codificado en base64, con el propósito de evitar la dependencia entre el servicio y el recurso. Sin embargo, estos SWA pueden contener recursos de gran tamaño que se ven afectados por su infraestructura y no pueden ser presentados correctamente, debido a un desbordamiento en el almacenamiento dinámico de la aplicación web. Es por ello que debe realizarse una estrategia de gestión que permita ejecutar SWA y presentar sus contenidos de manera correcta sin que se vean limitados por su tamaño.

El presente documento reporta la implementación de una estrategia de gestión a la para segmentar y reconstruir Recursos de Aprendizaje, solventando la problemática en el límite del almacenamiento dinámico que no permite presentar SWA con contenidos de gran tamaño. Las pruebas realizadas muestran la correcta segmentación, reconstrucción, generación, composición y presentación de SWA, tanto en servicios atómicos que solo contienen un recurso como para servicios compuestos que contienen al menos dos recursos de distintos tamaños. Los SWA utilizados en este trabajo fueron generados y probados en el protocolo SOAP y la arquitectura REST, con el propósito de comparar los tiempos de respuesta entre ellos.

Palabras clave: aprendizaje electrónico, base64, estrategia de gestión, reconstrucción, recursos de aprendizaje, segmentación, servicios web.

Abstract

The development of web applications has evolved over the years so that they are increasingly easier to use, but also that they increase their functions. Web services arise to offer certain advantages to the user, avoiding the dependence between modules or components of the applications.

The topics of applications and web services can be diverse, one of the most demanded areas at present is education where we can find too much educational content available on the Internet, also called Learning Resources. In this context, electronic learning (e-learning) is defined as the process of learning-teaching through the use of Information and Communication Technologies (ICTs). One way in which educational content is structured is through Learning Objects, which are composed of four pedagogical elements known as learning objective, informative content, learning activities and evaluation.

As a result of the above, the Learning Web Services (LWS) are created to contain some of these types of learning objects in a web service to be delivered to the user [Delgado Fernández, 2012]. These LWS contain a Learning Resource encoded in base64 to avoid dependence between service and resource. However, these LWS may contain large resources that are affected by their infrastructure and cannot be presented correctly, due to an overflow in the heap space of the web application. It's for this reason that a management strategy must be carried out that allows LWS to be executed and its contents to be presented correctly without being limited by their size.

This document reports the implementation of a management strategy to segment and rebuild Learning Resources, solving the problem in the limit of heap space that does not allow presenting LWS with large content. The tests performed show the correct generation, composition and presentation of LWS, both in atomic services that only contain one resource and for composite services that contain at least two resources of different sizes. The LWS used in this work were generated and tested in SOAP protocol and REST architecture to compare the response times between them.

Keywords: e-learning, base64, learning resources, management strategy, rebuilding, segmentation, web services.

Índice de Contenido

Lista de Acrónimos	v
Glosario de Términos.....	vii
Capítulo 1. Introducción.....	1
1.1. Presentación	2
1.2. Descripción del Problema	3
1.3. Justificación	3
1.4. Objetivos	4
1.4.1. Objetivo General.....	4
1.4.2. Objetivos Específicos	4
1.5. Alcances y Limitaciones	4
1.5.1. Alcances.....	4
1.5.2. Limitaciones	4
1.6. Beneficios	4
1.7. Organización del Documento.....	5
Capítulo 2. Marco Conceptual	6
2.1. Servicio Web.....	7
2.2. SWA: Servicio Web de Aprendizaje.....	7
2.3. Aprendizaje Electrónico (E-learning)	8
2.4. Sistemas Gestores	8
2.4.1. Sistema Gestor de Aprendizaje (Learning Management System).....	8
2.4.2. Sistema Gestor de Contenido (Content Management System)	9
2.5. OA: Objeto de Aprendizaje.....	9
2.6. RA: Recurso de Aprendizaje.....	9
2.7. SOA: Arquitectura Orientada a Servicios	10
2.8. SOAP: Protocolo de Acceso Simple al Objeto	11
2.9. REST: Transferencia de Estado Representacional	11
2.10. WSDL: Lenguaje de Descripción de Servicios Web	11
2.11. Esquema de Codificación Base64	12
2.12. Composición de Servicios Web	12
2.13. JVM: Máquina Virtual Java	13
2.13.1. Espacio de Almacenamiento Dinámico Java (Java Heap Space).....	14
2.14. Estrategia de Gestión.....	14
2.15. Segmentación	14
2.16. Reconstrucción.....	16

Capítulo 3. Antecedentes y Trabajos Relacionados	17
3.1. Antecedentes de la Investigación	18
3.2. Trabajos Relacionados	21
3.2.1. Trabajos de Estrategias de Gestión	21
3.2.2. Trabajos de Generación y Composición de Servicios Web	24
3.3. Resumen Comparativo de los Trabajos Relacionados	27
3.3.1. Comparación de los Trabajos de Estrategias de Gestión	27
3.3.2. Comparación de los Trabajos de Generación y Composición de Servicios Web	29
Capítulo 4. Metodología de Solución	32
4.1. Metodología de Solución	33
I. Análisis de los procesos de generación y ejecución de SWA con contenido	33
II. Definición de la Estrategia de Gestión	34
III. Implementación de la Estrategia de Gestión	35
Capítulo 5. Pruebas y Resultados	47
5.1. Pruebas Preliminares	48
5.2. Ejecución de Pruebas	50
5.2.1. Objetivos de las Pruebas	50
5.3. Procedimiento de Pruebas	50
I. Descripción de los Elementos de Prueba	51
II. Descripción de las Interfaces	54
III. Ejecución de las Pruebas	56
5.4. Resultados y Conclusiones de las Pruebas	61
Capítulo 6. Conclusiones y Trabajo Futuro.....	64
6.1. Conclusiones	65
6.1.1. Objetivos Cumplidos	65
6.1.2. Conclusiones Específicas	65
6.2. Aportaciones	67
6.3. Trabajos Futuros	68
Referencias.....	69
Anexo A. Descripción de las Pruebas Preliminares	74
1 ^{ra} Prueba Preliminar: “Segmentación y reconstrucción de recursos contenidos en Servicios Web de Aprendizaje mediante segmentos de distintos tamaños”	75
2 ^{da} Prueba Preliminar: “Generación de Servicios Web de Aprendizaje para casos de prueba usando las actividades de la estrategia”	78
Anexo B. Descripción de las Interfaces	80
Interfaces de la herramienta Java: “Generación y Composición”	81
Interfaces del cliente PHP: “Reconstrucción y Presentación”	83

Anexo C. Descripción de los Tipos de Pruebas.....	88
1 ^{er} Tipo de Prueba: “Segmentación de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje”	89
2 ^{do} Tipo de Prueba: “Presentación de Servicios Web de Aprendizaje con contenido segmentado”	92
3 ^{er} Tipo de Prueba: “Composición de Servicios Web de Aprendizaje con contenidos segmentados”	95
4 ^{to} Tipo de Prueba: “Presentación de Servicios Web de Aprendizaje Compuestos con contenidos segmentados”	97
Anexo D. Descripción de las Pruebas Adicionales.....	100
Comportamiento de los Recursos de Aprendizaje en distintos formatos	101

Índice de Figuras

Figura 2.1. Servicio Web de Aprendizaje como intermediario [Delgado Fernández, 2012].....	7
Figura 2.2. Servicio Web de Aprendizaje con contenido [Salinas Roman, 2017].....	8
Figura 2.3. Objeto de Aprendizaje [Sicilia, 2016].....	9
Figura 2.4. Arquitectura Orientada a Servicios [Alfonso, 2012].....	10
Figura 2.5. Codificación y decodificación en Base64	12
Figura 2.6. Composición de Servicios Web de Aprendizaje [Escobar Megchún, 2017].....	13
Figura 2.7. Arquitectura de la Máquina Virtual Java (JVM).....	13
Figura 2.8. Segmentación en tiempo de carga	15
Figura 2.9. Segmentación en tiempo de ejecución	15
Figura 2.10. Segmentación de un recurso.....	16
Figura 2.11. Reconstrucción de un recurso segmentado	16
Figura 3.1. Antecedentes relacionados a los Servicios Web de Aprendizaje	18
Figura 4.1. Metodología de solución	33
Figura 4.2. Proceso de generación de un SWA en la herramienta antecedente [Escobar Megchún, 2017].....	34
Figura 4.3. Diagrama de casos de uso	34
Figura 4.4. Proceso de generación de SWA con la estrategia de gestión	35
Figura 4.5. Diagrama de clases de la herramienta extendida.....	36
Figura 4.6. Definición del límite de tamaño para realizar la segmentación.....	36
Figura 4.7. Código implementando la condición de segmentación	37
Figura 4.8. Modificación al WSDL de un SWA como actividad de reconstrucción	37
Figura 4.9. WSDL de un SWA con contenido segmentado	38
Figura 4.10. Interfaz de un Servicio Web de Aprendizaje en SOAP.....	39
Figura 4.11. Métodos de implementación para un SWA en SOAP.....	39
Figura 4.12. Segmento del recurso de aprendizaje contenido en un SWA en SOAP	40
Figura 4.13. Implementación de un Servicio Web de Aprendizaje en REST.....	40
Figura 4.14. Archivo de configuración para el despliegue de un SWA en REST	41
Figura 4.15. Segmento del recurso de aprendizaje contenido en un SWA en REST	41
Figura 4.16. Diagrama BPMN para la generación de un SWA implementando la estrategia de gestión	42
Figura 4.17. Diagrama de secuencia para generar y presentar un SWA mediante la estrategia de gestión.....	43
Figura 4.18. Proceso de composición de SWA en la herramienta antecedente [Escobar Megchún, 2017].....	44
Figura 4.19. Reglas de composición [Escobar Megchún, 2017]	44

Figura 4.20. Proceso de composición de SWA mediante separación de protocolos	45
Figura 4.21. Diagrama BPMN para la composición de SWA separándolos por sus protocolos	45
Figura 5.1. Lista de pruebas preliminares realizadas.....	48
Figura 5.2. Reconstrucción no exitosa de un recurso segmentado con código blanco	48
Figura 5.3. Comparación en los tiempos de respuesta en los rangos de reconstrucción	49
Figura 5.4. Procedimiento de pruebas y resultados	51
Figura 5.5. Elementos de prueba	51
Figura 5.6. Entorno de pruebas Apache JMeter	53
Figura 5.7. Mapas de navegación para las interfaces de la herramienta Java y el cliente PHP	54
Figura 5.8. Lista de tipos de pruebas realizadas	56
Figura 5.9. Comparación en los tiempos de segmentación de RA para generar SWA en SOAP y REST	57
Figura 5.10. Comparación en los tiempos de reconstrucción de RA empaquetados como SWA	58
Figura 5.11. Reconstrucción y presentación de un SWA	58
Figura 5.12. Comparación en los tiempos de presentación de SWA-C en SOAP y REST	59
Figura 5.13. Presentación exitosa de un Servicio Web de Aprendizaje Compuesto	60
Figura B.1. Vista inicial de la herramienta Java.....	81
Figura B.2. Vista para la generación de un SWA mediante la estrategia de gestión	82
Figura B.3. Vista para la composición de SWA con contenidos segmentados	83
Figura B.4. Vista inicial del cliente PHP.....	84
Figura B.5. Vista para la presentación de un SWA con contenido segmentado en SOAP	85
Figura B.6. Vista para la presentación de un SWA con contenido segmentado en REST	85
Figura B.7. Vista para la presentación de un SWA-C con contenidos segmentados en SOAP.....	86
Figura B.8. Vista para la presentación de un SWA-C con contenidos segmentados en REST	87
Figura D.1. Comportamiento de RA en formato MP4	101
Figura D.2. Comportamiento de RA en formato PDF.....	102
Figura D.3. Comportamiento de RA en formato DOC.....	102
Figura D.4. Comportamiento de RA en formato PPT	103

Índice de Tablas

Tabla 3.1. Resumen comparativo de trabajos relacionados a estrategias de gestión	27
Tabla 3.2. Resumen comparativo de trabajos relacionados a generación y composición de servicios web	29
Tabla 5.1. Repositorios de recursos.....	52
Tabla 5.2. Lista de recursos de aprendizaje seleccionados	52
Tabla 6.1. Objetivos cumplidos.....	65
Tabla A.1. Datos utilizados en la Prueba Preliminar 1	75
Tabla A.2. Rangos probados para definir el tamaño en la actividad de segmentación	76
Tabla A.3. Doble conversión de una cadena ASCII a binario, y de binario a Base64.....	76
Tabla A.4. Resultados en los tiempos de reconstrucción para segmentos de distintos tamaños.....	77
Tabla A.5. Datos utilizados en la Prueba Preliminar 2	78
Tabla A.6. Resultados de generación aplicando la actividad de segmentación	79
Tabla C.1. Datos utilizados en el Tipo de Prueba 1.....	89
Tabla C.2. Resultados de segmentación de RA para generar SWA en SOAP y REST	90
Tabla C.3. Datos utilizados en el Tipo de Prueba 2.....	92
Tabla C.4. Resultados de reconstrucción y presentación con SOAP y REST	93
Tabla C.5. Datos utilizados en el Tipo de Prueba 3.....	95

Tabla C.6. Resultados de composición para SWA con contenidos segmentados.....	96
Tabla C.7. Lista de SWA-C generados en SOAP y REST	96
Tabla C.8. Datos utilizados en el Tipo de Prueba 4.....	97
Tabla C.9. Resultados de presentación para SWA-C con contenidos segmentados.....	98

Lista de Acrónimos

API	Interfaz de Programación de Aplicaciones
ASCII	Código Estándar Americano para el Intercambio de Información
Base64	Sistema de Codificación de Transmisión de contenido MIME
BPMN	Modelo y Notación de Proceso de Negocios
CMS	Sistema Gestor de Contenido
DII	Interfaz de Innovación Dinámica
E-Learning	Aprendizaje Electrónico
HTTP	Protocolo de Transferencia de Hipertexto
HTML	Lenguaje de Marcas de Hipertexto
JVM	Máquina Virtual Java
LMS	Sistema Gestor de Aprendizaje
MIME	Extensiones Multipropósito de Correo de Internet
OA	Objeto de Aprendizaje
QoS	Calidad de Servicio
RA	Recurso de Aprendizaje
REST	Transferencia de Estado Representacional
SOA	Arquitectura Orientada a Servicios
SOAP	Protocolo de Acceso Simple al Objeto
SW	Servicio Web
SWA	Servicio Web de Aprendizaje
SWA-C	Servicio Web de Aprendizaje Compuesto
TICs	Tecnologías de Información y Comunicación
UDDI	Descripción Universal, Descubrimiento e Integración
URI	Identificador de Recursos Uniforme
W3C	Consortio Mundial de la Red
WS-BPEL	Lenguaje de Ejecución de Procesos de Negocios con Servicios Web
WSDL	Lenguaje Descriptor del Servicio Web
XML	Lenguaje de Marcado Extensible

Glosario de Términos

Base64: es un esquema de codificación de binario a texto, es decir, es un tipo de codificación de transporte de datos que tiene la finalidad de proteger la información que se envía entre dos o más ordenadores. Esta codificación se hace representando los datos binarios mediante una cadena ASCII traduciéndolos en una representación radix-64. Para asegurar que los datos se mantengan intactos y sin modificaciones durante la transmisión. Es usado en muchas aplicaciones, incluyendo la escritura de emails vía MIME y el almacenamiento de datos complejos en XML [MDN Web Docs, 2013].

Lenguaje Descriptor del Servicio Web (WSDL): es la capa de descripción de servicio dentro del protocolo de servicios web, es una gramática XML para especificar una interfaz pública para un servicio web. Esta interfaz puede incluir información de sus funciones disponibles, tipos de datos entre los mensajes XML, protocolos de transporte específicos y las direcciones para localizar al servicio especificado [Cerami, 2002].

Protocolo de Acceso Simple al Objeto (SOAP): es un marco estándar, extensible y componible para empaquetar e intercambiar mensajes XML a través de una variedad de protocolos subyacentes. SOAP proporciona un mecanismo conveniente para referenciar sus capacidades el cual ha sido diseñado para ser independiente a cualquier modelo de programación y de implementaciones semánticas específicas [W3C Working Group, 2004].

Recurso de Aprendizaje (RA): es una entidad digital que puede usarse, reutilizarse o referenciarse durante el proceso de aprendizaje a través de las TICs. Estos son creados con materiales básico, herramientas, metodologías o capacitaciones que involucren facilitar el proceso de enseñanza-aprendizaje al lector, desde temas sencillos como aprender a sumar y restar, hasta temas complejos como aprender a programar software o realizar cirugías quirúrgicas [Van Assche & Vuorikari, 2006].

Servicio Web de Aprendizaje (SWA): es un recurso de aprendizaje en término de servicio web que contiene el mecanismo que comunica el contenido educativo de uno de los cuatro elementos que conforman a un objeto de aprendizaje, funcionando como un intermediario entre el usuario y el contenido de aprendizaje [Delgado Fernández, 2012].

Transferencia de Estado Representacional (REST): es un estilo de arquitectura que define un conjunto restricciones y propiedades basadas en HTTP. REST es usado en conjunción con un Identificador de Recursos Uniforme (URI) y HTTP, donde se considera a un servicio web en REST como un recurso que puede ser identificado y localizado por una URI, y a su vez, diferentes operaciones pueden ser ejecutados sobre los recursos mediante el uso de métodos HTTP. REST es un principio de diseño para aplicaciones web basadas en la arquitectura cliente-servidor con la finalidad de usar la mínima cantidad de métodos HTTP [Potti, Ahuja, Umapathy, & Prodanoff, 2012].

Capítulo

1

Introducción

En este capítulo se describe el planteamiento del problema que atiende esta investigación, la justificación, el objetivo general y sus objetivos específicos, los alcances y las limitaciones, los beneficios y la organización del documento de tesis.

1.1. Presentación

Con el paso de los años el Internet se ha vuelto algo tan común en nuestra vida cotidiana debido a que es el medio más usado que nos permite comunicarnos, conocer las noticias más recientes de cualquier parte del mundo, ver o escuchar contenido multimedia para entretenernos, consultar indicaciones para llegar a algún destino, realizar compras y pedidos a domicilio, hacer reservaciones a eventos o lugares, y entre muchas otras funciones que nos ofrece el Internet se encuentra el aprendizaje electrónico.

El aprendizaje electrónico, también conocido como e-learning es el uso de las Tecnologías de Información y Comunicación (TICs) para facilitar el proceso de enseñanza y aprendizaje, proporcionando Sistemas Gestores de Aprendizaje (LMS, por sus siglas en inglés "Learning Management System") que ofrecen cursos con contenidos sobre temas específicos, estas plataformas educativas permiten administrar, controlar y distribuir las actividades de formación educativa en línea. Una de las formas en que nos brindan esta formación es mediante Objetos de Aprendizaje, los cuales son entidades digitales con contenidos educativos que presentan Recursos de Aprendizaje de manera visual al usuario.

Un Objeto de Aprendizaje está compuesto de cuatro elementos pedagógicos conocidos como: objetivo de aprendizaje, contenido informativo, actividades de aprendizaje y evaluación. Mientras que un Recurso de Aprendizaje es un archivo de texto, audio, imagen o vídeo con contenido educativo que puede ser visualizado por una aplicación [Sicilia, 2016].

Una manera para aumentar la reusabilidad y la interoperabilidad del material educativo entre plataformas es a través del desarrollo de servicios web. Se define como Servicio Web de Aprendizaje al intermediario que contiene un Recurso de Aprendizaje que es entregado a un cliente a través de Internet [Delgado Fernández, 2012].

Con el propósito de evitar la dependencia entre el servicio y el recurso, se denomina como Servicio Web de Aprendizaje con contenido, cuando el servicio contiene de manera codificada al Recurso de Aprendizaje que se entrega al cliente. Estos servicios empaquetan al recurso en el esquema de codificación Base64 [Salinas Roman, 2017].

El trabajo de [Escobar Megchún, 2017] presenta una herramienta para generar y componer Servicios Web de Aprendizaje a partir de recursos con contenido educativo. Donde se toman Recursos de Aprendizaje y son codificados en Base64 para ser empaquetados como servicios web. Sin embargo, aún existen problemas al momento de ejecutar estos Servicios Web de Aprendizaje debido a que el contenido es de gran tamaño y supera el límite en el almacenamiento dinámico de la aplicación, lo que ocasiona que el recurso no pueda ser presentado al cliente. Y por tal motivo, el usuario se queda sin poder visualizar el contenido educativo.

En este trabajo de investigación se plantea determinar una estrategia de gestión de Recursos de Aprendizaje que permita segmentar y reconstruir de manera automática Servicios Web de Aprendizaje con contenidos de gran tamaño. Con el propósito de presentar el material educativo correctamente, sin verse afectado por su tamaño.

En este contexto, se propone una estrategia de gestión capaz de segmentar Recursos de Aprendizaje contenidos en Servicio Web de Aprendizaje y reconstruirlos de manera automática para poder ser entregados al cliente, además de permitir la presentación de Servicios Web de Aprendizaje Compuestos con contenidos de distintos tamaños.

1.2. Descripción del Problema

Los Servicios Web de Aprendizaje con contenido, son servicios web que contienen Recursos de Aprendizaje codificados para ser entregados a usuarios. Este tipo de servicios se construye para evitar dependencias de los servicios hacia los recursos. Sin embargo, su uso puede verse limitado por la infraestructura que los soporta. Por lo tanto, es necesario segmentar los recursos, cuando sea requerido, para ser presentados a los usuarios sin verse afectados por su tamaño. El problema de segmentar los recursos es que debe mantenerse una secuencia que permita reconstruirlos correctamente cuando se quieran ejecutar, y no sean llamados en un orden indistinto que pueda dañar su contenido, por lo que se debe obedecer a una guía o estrategia de gestión.

1.3. Justificación

El uso de plataformas educativas ha incrementado de manera gradual con el paso de los años debido a que la tecnología es cada vez más común en nuestra vida cotidiana, lo que ocasiona que gran parte de la población utilice el aprendizaje electrónico como medio principal a la hora de obtener información o conocimientos sobre algún tema en particular.

A la fecha, los servicios web tienen gran tendencia en reusabilidad de componentes de software, ya que brindan distintas funcionalidades al usuario. Entre estas se encuentran los Servicios Web de Aprendizaje que contienen recursos con material educativo. Sin embargo, cuando un Recurso de Aprendizaje es demasiado grande como en el caso de vídeos o documentos con muchas imágenes o audios, no pueden ser soportados ya que rebasan el límite del almacenamiento dinámico. Por lo que es necesario definir una estrategia de gestión para los Servicios Web de Aprendizaje con contenido, la cual nos permita presentar correctamente Recursos de Aprendizaje de gran tamaño.

El propósito de este trabajo es implementar dicha estrategia de gestión a la herramienta Java “LWS” para la generación y composición de Servicios Web de Aprendizaje desarrollada por [Escobar Megchún, 2017], para solventar el problema latente en el límite del almacenamiento dinámico que no permite presentar servicios con contenidos de gran tamaño, los cuales superan los 50 MB. Esto ocasiona que no puedan visualizarse de manera correcta todos los Recursos de Aprendizaje al usuario.

1.4. Objetivos

1.4.1. Objetivo General

Definir una estrategia de gestión de Recursos de Aprendizaje para que puedan participar en procesos de generación de Servicios Web de Aprendizaje con contenidos de gran tamaño.

1.4.2. Objetivos Específicos

- Conocer las características para una guía de reconstrucción automática.
- Realizar la composición de servicios web cuyos contenidos fueron segmentados para gestionar su contenido y presentarlo de manera correcta.

1.5. Alcances y Limitaciones

Para propósitos de este trabajo, se establecen los alcances y limitaciones como se describen a continuación.

1.5.1. Alcances

- Identificar una forma de segmentar un Recurso de Aprendizaje para que pueda ser contenido en un Servicio Web de Aprendizaje.
- Identificar la estrategia en que deben componerse los Servicios Web de Aprendizaje con contenidos que han sido segmentados.

1.5.2. Limitaciones

Para propósitos de pruebas, únicamente se considera un conjunto de Recursos de Aprendizaje con los formatos de los siguientes tipos:

- *Audio:* MP3.
- *Imagen:* JPG, PNG.
- *Texto:* DOC, PDF, PPT.
- *Vídeo:* MP4.

Es decir, no se consideraron todos los tipos de Recursos de Aprendizaje.

1.6. Beneficios

El principal beneficio del presente trabajo es contar con una estrategia de gestión que permita segmentar Recursos de Aprendizaje, para reconstruirlos y presentarlos directamente en la vista del cliente. Con el fin de solventar los problemas en el almacenamiento dinámico, proporcionando la posibilidad de tener recursos de mayores tamaños para aumentar las

capacidades del contenido informativo que pueden ofrecer las plataformas educativas a través de Servicios Web de Aprendizaje.

1.7. Organización del Documento

La organización del presente documento y la distribución de sus capítulos, así como una breve descripción de los mismos, son presentados a continuación:

- **Capítulo 2 ‘Marco Conceptual’:** Presenta el contexto teórico del tema a tratar en este trabajo de investigación, donde se identifican y describen los principales conceptos que el lector debe conocer.
- **Capítulo 3 ‘Antecedentes y Trabajos Relacionados’:** Respalda al presente tema de investigación mediante trabajos realizados antes del propuesto, así como trabajos que lo han precedido, dando sustento al surgimiento de este trabajo.
- **Capítulo 4 ‘Metodología de Solución’:** Describe la metodología empleada para definir e implementar la estrategia de gestión para los Servicios Web de Aprendizaje con contenido, así como las actividades necesarias para llevar a cabo los procesos de segmentación, reconstrucción y composición.
- **Capítulo 5 ‘Pruebas y Resultados’:** Documenta las pruebas realizadas al producto desarrollado en la metodología de solución, para probar la correcta aplicación de la estrategia de gestión para los Servicios Web de Aprendizaje y se realiza un análisis de los resultados obtenidos.
- **Capítulo 6 ‘Conclusiones y Trabajo Futuro’:** Presenta la conclusión alcanzada en el presente trabajo de investigación con base a la metodología implementada y los resultados obtenidos, así como propone posibles trabajos futuros a raíz de este trabajo.

Capítulo

2

Marco Conceptual

En este capítulo se presentan los conceptos y elementos teóricos necesarios para comprender el contexto de esta investigación. Se describen algunos tales como: Objeto de Aprendizaje, Recurso de Aprendizaje, Transferencia de Estado Representacional (REST), Servicio Web, Servicio Web de Aprendizaje, Protocolo de Acceso Simple al Objeto (SOAP), entre otros.

2.1. Servicio Web

Un Servicio Web es una interfaz que permite acceder a las funcionalidades de una aplicación a través de la red, construido mediante tecnologías estándar de Internet.

“Un servicio Web es un sistema de software diseñado para apoyar la interacción interoperable máquina a máquina sobre una red. Tiene una interfaz descrita en un formato procesable por una máquina (específicamente WSDL). Otros sistemas interactúan con el servicio Web de una manera prescrita por su descripción utilizando mensajes SOAP típicamente transportados usando HTTP con una serialización XML en conjunto con otras normas relacionadas con la Web” [W3C Working Group, 2004].

Los servicios web proporcionan estándares de comunicación entre las diversas aplicaciones de software que se encuentran en la presentación de información como contexto dinámico orientada al usuario.

2.2. SWA: Servicio Web de Aprendizaje

Un Servicio Web de Aprendizaje (SWA) es un recurso de aprendizaje en términos de servicio web que contiene el mecanismo que comunica el contenido educativo de uno de los cuatro elementos que conforman a un objeto de aprendizaje, funcionando como un intermediario entre el usuario y el contenido de aprendizaje [Delgado Fernández, 2012].

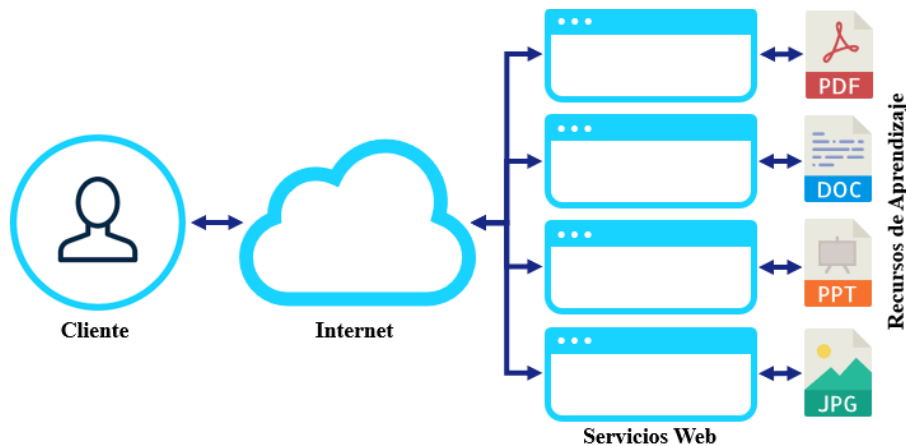


Figura 2.1. Servicio Web de Aprendizaje como intermediario [Delgado Fernández, 2012]

Un SWA potencializa las características de un objeto de aprendizaje reduciendo sus limitantes de uso, actualización, disponibilidad y creación dentro de un sistema gestor de aprendizaje, lo que facilita su utilización, reutilización, interoperabilidad y accesibilidad al contenido educativo.

Se considera a un SWA como auto contenido cuando el servicio contiene de manera codificada al recurso de aprendizaje que se entrega a un cliente a través de Internet. Estos son construidos para evitar dependencias entre servicios y los recursos [Salinas Roman, 2017].



Figura 2.2. Servicio Web de Aprendizaje con contenido [Salinas Roman, 2017]

Para propósitos de este trabajo de investigación se utiliza el término de Servicios Web de Aprendizaje con contenido para evitar el uso en la denominación auto contenido, ya que podría considerarse como un concepto más abstracto o complejo para el lector.

2.3. Aprendizaje Electrónico (E-learning)

Se define al Aprendizaje Electrónico, del término en inglés E-learning, como “*la convergencia de Internet y el aprendizaje; el uso de las tecnologías Web para crear, promover, entregar y facilitar el aprendizaje en cualquier momento y en cualquier lugar*” [Madjarov & Boucelma, 2006].

El aprendizaje electrónico es definido desde el punto de vista educacional y usado en las Tecnologías de Información y Comunicación (TICs), las cuales permiten la gestión del aprendizaje facilitando el proceso de enseñanza-aprendizaje a través de la comunicación, colaboración, administración y herramientas para la presentación de información; además de facilitar la gestión, integración y distribución de los Objetos de Aprendizaje, que son la unidad fundamental en los Sistemas Gestores de Aprendizaje.

2.4. Sistemas Gestores

Existe una gran variedad de sistemas gestores que nos permiten controlar y administrar el contenido de información dentro de una aplicación, como es el caso de las plataformas educativas llamadas Sistemas Gestores de Aprendizaje, algunos ejemplos de ellos son Moodle, Dokeos, Edmodo, Udemy, entre muchos otros. También encontramos a los Sistemas Gestores de Contenido encargados de controlar los recursos o la clasificación de su información, un ejemplo de estos son los objetos de aprendizaje que dividen en elementos a los recursos de aprendizaje de acuerdo a su tipo de contenido.

2.4.1. Sistema Gestor de Aprendizaje (Learning Management System)

Un Sistema Gestor de Aprendizaje, del término en inglés Learning Management System (LMS), maneja los aspectos del proceso de aprendizaje; un LMS entrega y administra contenido instruccional, identifica y evalúa el aprendizaje organizacional e individual, da seguimiento a los progresos para cumplir con los objetivos, reúne y presenta datos para la supervisión de su proceso de aprendizaje [Szabo & Flesher, 2002].

2.4.2. Sistema Gestor de Contenido (Content Management System)

Un Sistema Gestor de Contenido, del término en inglés Content Management System (CMS), es usado para crear, almacenar, ensamblar y entregar aprendizaje electrónico personalizado en forma de objetos de aprendizaje. Definido como “una aplicación de software (o conjunto de aplicaciones) que gestionan la creación, almacenamiento, uso y reusabilidad de objetos de aprendizaje” [Friesen, 2005].

2.5. OA: Objeto de Aprendizaje

Un Objeto de Aprendizaje (OA) es una entidad digital distribuida por Internet para ser consultada y utilizada por diversos usuarios simultáneamente. Se define como una unidad didáctica de aprendizaje en formato digital, independiente, auto contenido, perdurable y predispuesta para su reutilización en varios contextos educativos por la inclusión de información auto descriptiva en forma de metadatos [Sicilia, 2016].

Un Objeto de Aprendizaje se encuentra conformado de cuatro elementos pedagógicos que se encuentran unidos a través de un metadato.

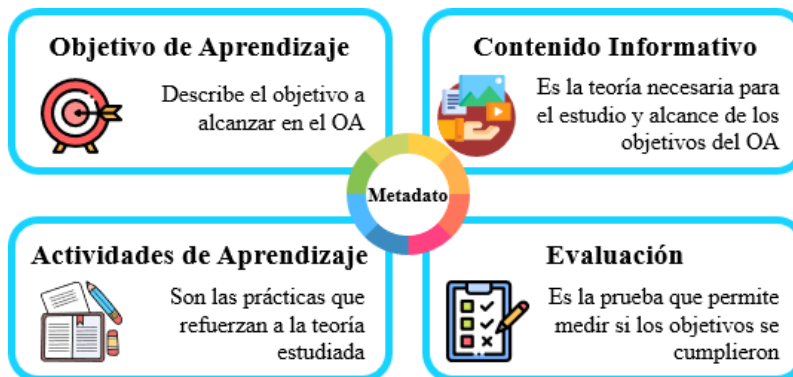


Figura 2.3. Objeto de Aprendizaje [Sicilia, 2016]

2.6. RA: Recurso de Aprendizaje

Un Recurso de Aprendizaje (RA) es una entidad digital que puede usarse, reutilizarse o referenciarse durante el proceso de aprendizaje a través de las TICs. Estos son creados con materiales básico, herramientas, metodologías o capacitaciones que involucren facilitar el proceso de enseñanza-aprendizaje al lector, desde temas sencillos como aprender a sumar y restar, hasta temas complejos como aprender a programar software o realizar cirugías quirúrgicas [Van Assche & Vuorikari, 2006].

En otras palabras, un Recurso de Aprendizaje puede ser cualquier archivo con contenido educativo capaz de ser visualizado por una aplicación.

2.7. SOA: Arquitectura Orientada a Servicios

Una Arquitectura Orientada a Servicios (SOA) se define como “un término que representa un modelo en el que la lógica de automatización se descompone en unidades de lógica más pequeñas y distintas. En conjunto, estas unidades comprenden una parte más grande de la lógica de automatización de negocios. Individualmente, estas unidades se pueden distribuir” [Erl, 2005].

SOA puede referirse a la arquitectura de una aplicación o al enfoque utilizado para estandarizar la arquitectura de una empresa, es un método mediante el cual distintos tipos de servicios pueden interactuar independientemente entre sí.

Los servicios web interactúan como una SOA capaz de crear una definición abstracta del servicio, proporcionar una implementación concreta de éste, publicarlo, localizarlo, seleccionar una instancia y utilizarla con una elevada interoperabilidad. Cuyas implementaciones se encuentran a nivel de lógica y transporte, las cuales estructuran la SOA del servicio web [Alfonso, 2012].

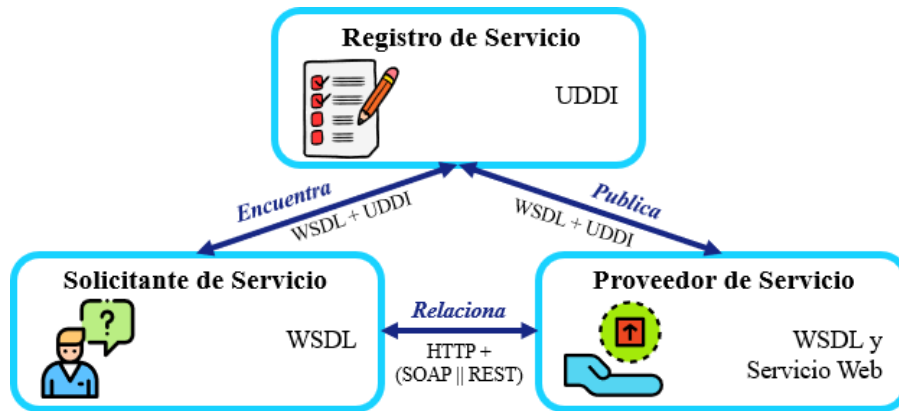


Figura 2.4. Arquitectura Orientada a Servicios [Alfonso, 2012]

Donde las implementaciones lógicas son los roles y las de transporte son las operaciones que la componen:

- **Registro de Servicio**, es un repositorio de servicios que permite la búsqueda en sus registros hacia sus proveedores.
- **Solicitante de Servicio**, es una aplicación que busca en el registro de servicio a través de una petición al servicio que cumpla con sus necesidades.
- **Proveedor de Servicio**, es una entidad que válida y permite las peticiones hacia sus solicitantes. Publica una descripción del servicio que éste ofrece vía registro de servicio.

2.8. SOAP: Protocolo de Acceso Simple al Objeto

El Protocolo de Acceso Simple al Objeto (SOAP) es un marco estándar, extensible y componible para empaquetar e intercambiar mensajes XML a través de una variedad de protocolos subyacentes. En este contexto, SOAP proporciona un mecanismo conveniente para referenciar sus capacidades el cual ha sido diseñado para ser independiente a cualquier modelo de programación y de implementaciones semánticas específicas [W3C Working Group, 2004].

2.9. REST: Transferencia de Estado Representacional

La Transferencia de Estado Representacional (REST) es un estilo de arquitectura que define un conjunto restricciones y propiedades basadas en HTTP. La arquitectura REST es utilizado en conjunción con un Identificador de Recursos Uniforme (URI) y HTTP, donde se considera a un servicio web en REST como un recurso que puede ser identificado y localizado por una URI, y a su vez, diferentes operaciones pueden ser ejecutados sobre los recursos mediante el uso de métodos HTTP. REST es un principio de diseño para aplicaciones web basadas en la arquitectura cliente-servidor con la finalidad de usar la mínima cantidad de métodos HTTP [Potti, Ahuja, Umapathy, & Prodanoff, 2012].

Los servicios web de tipo REST suelen integrarse mejor con HTTP que los servicios basados en SOAP, ya que no requieren mensajes XML o definiciones del servicio en forma de fichero WSDL. Tienen una infraestructura “ligera” que permite que los servicios se construyan utilizando herramientas de forma mínima [Alfonso, 2012].

2.10. WSDL: Lenguaje de Descripción de Servicios Web

El Lenguaje de Descripción de Servicios Web (WSDL) es *“un formato XML para describir servicios de red como un conjunto de puntos finales que operan en mensajes que contienen información orientada a documentos u orientada a procedimientos. Las operaciones y los mensajes se describen de forma abstracta y, a continuación, se vinculan a un protocolo de red concreto y a un formato de mensaje para definir un punto final. Los puntos finales concretos relacionados se combinan en puntos finales abstractos (servicios). WSDL es extensible para permitir la descripción de los puntos finales y sus mensajes, independientemente de qué formatos de mensaje o protocolos de red se utilizan para comunicarse”* [W3C Working Group, 2004].

WSDL se define como la capa de descripción de servicio dentro del protocolo de servicios web, es una gramática XML para especificar una interfaz pública para un servicio web. Esta interfaz puede incluir información de sus funciones disponibles, tipos de datos entre los mensajes XML, protocolos de transporte específicos y las direcciones para localizar al servicio especificado [Cerami, 2002].

2.11. Esquema de Codificación Base64

El esquema de codificación Base64 es un grupo de esquemas de codificación de binario a texto, es decir, es un tipo de codificación de transporte de datos que tiene la finalidad de proteger la información que se envía entre dos o más ordenadores. Esta codificación se hace representando los datos binarios mediante una cadena ASCII traduciéndolos en una representación radix-64. Cuya finalidad es asegurar que los datos se mantengan intactos y sin modificaciones durante la transmisión. Comúnmente es usado en muchas aplicaciones, incluyendo la escritura de emails vía MIME y el almacenamiento de datos complejos en XML [MDN Web Docs, 2013].

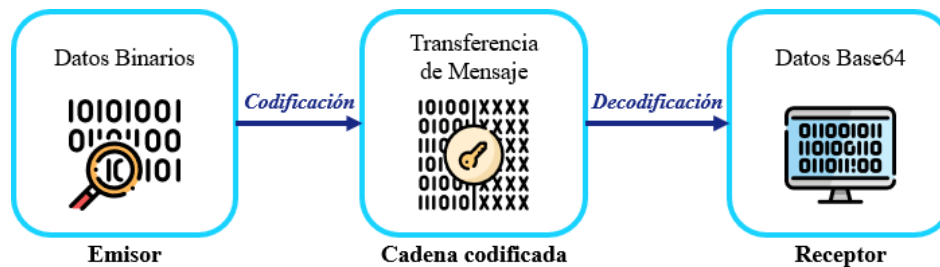


Figura 2.5. Codificación y decodificación en Base64

La codificación consiste en que el emisor convierte un mensaje en una cadena de caracteres que pueden ser recibidos y entendidos por un receptor, y la decodificación consiste en que el receptor convierte la cadena de caracteres que recibe en un mensaje, para traducir y entender el contenido que es entregado por el emisor.

En este trabajo, un SWA (emisor) tiene un recurso codificado en Base64 (mensaje) para ser enviado al cliente (receptor) mediante un protocolo de transferencia de datos, de tal forma que la aplicación web pueda ejecutar y presentar al recurso de aprendizaje.

2.12. Composición de Servicios Web

La composición de servicios web consiste en crear nuevos servicios con la combinación de servicios existentes. Estos servicios requieren ser compuestos para que servicios pequeños sean combinados como un servicio más grande que proporcione un valor específico. La composición de servicios web es el proceso de agregar múltiples servicios en un solo servicio para realizar funciones más complejas [Sheng; et. al., 2014].

Muchos enfoques de composición apuntan a automatizar la composición, lo que promete un desarrollo de aplicaciones más rápido, una reutilización más segura, y facilitar la interacción del usuario con conjuntos de servicios complejos. Con la composición automática, el usuario final o el desarrollador de la aplicación específica un objetivo (expresado en un lenguaje de descripción o notación matemática) y un motor de composición “inteligente” que selecciona los servicios adecuados y ofrece la composición de forma transparente al usuario. Los principales problemas son cómo identificar a los servicios candidatos y verificar qué tan cerca coinciden con una solicitud [Milanovic & Malek, 2004].

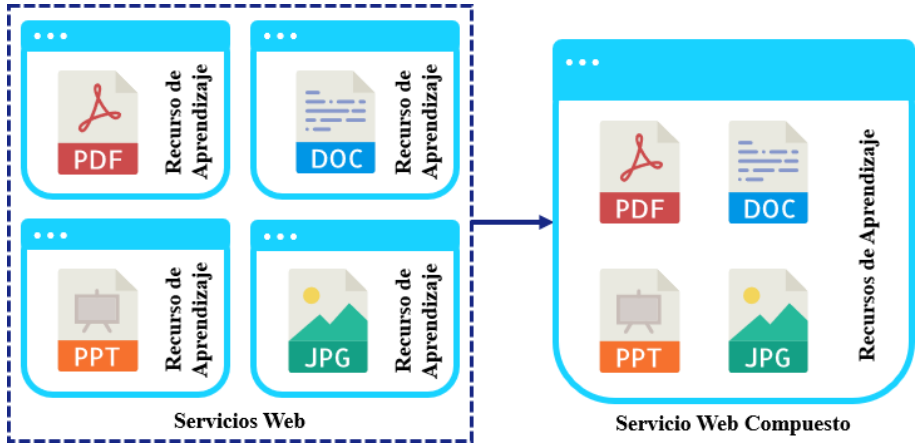


Figura 2.6. Composición de Servicios Web de Aprendizaje [Escobar Megchún, 2017]

La composición de Servicios Web de Aprendizaje con contenidos consiste en establecer un arreglo que permita invocar múltiples Recursos de Aprendizaje que serán entregados al cliente en una sola vista, generando un Servicio Web de Aprendizaje Compuesto (SWA-C).

2.13. JVM: Máquina Virtual Java

La Máquina Virtual Java, del término en inglés Java Virtual Machine (JVM), es una instancia virtual del motor de ejecución que ejecuta los códigos de bytes en archivos de clases Java sobre un microprocesador [Oracle Docs, 2016].

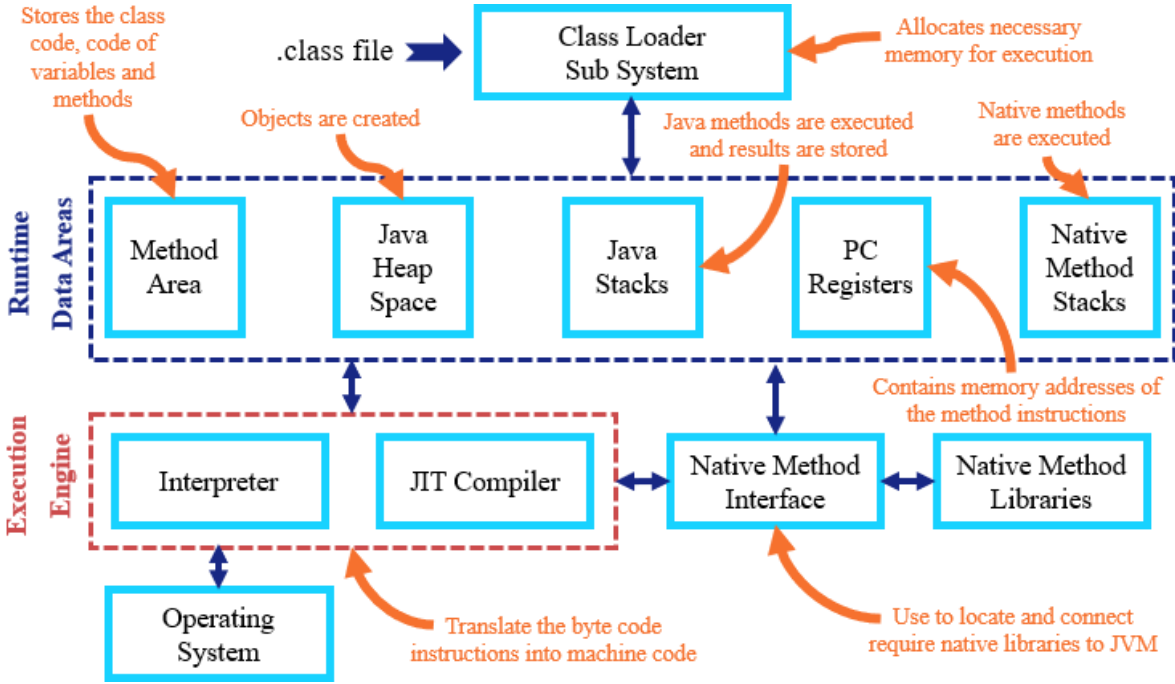


Figura 2.7. Arquitectura de la Máquina Virtual Java (JVM)

Al ejecutar un programa “.java” se convierte a un archivo “.class” que consta de instrucciones de códigos de bytes por el compilador Java. Ahora el archivo “.class” se entrega a la JVM.

2.13.1. Espacio de Almacenamiento Dinámico Java (Java Heap Space)

Los objetos Java residen en un área llamada Espacio de Almacenamiento Dinámico (Heap Space). Este espacio de almacenamiento es creado al inicio de la Máquina Virtual Java (JVM) y puede aumentar o disminuir de tamaño mientras se ejecuta la aplicación. Cada vez que la JVM carga el archivo “.class”, se crea un método y un Espacio de Almacenamiento Dinámico inmediatamente en la JVM [Oracle Docs, 2016].

2.14. Estrategia de Gestión

En la vida diaria, el término estrategia se usa para demostrar que una persona es capaz de pensar cada paso que da, de tal manera que dichas acciones vayan concatenadas y tengan relación. Algunas personas relacionan estrategia con la astucia y la malicia que cada ser humano posee como don natural para lograr algo [Contreras Sierra, 2013].

Se debe demostrar la importancia que se tiene al conocer con exactitud todas las actividades involucradas en una estrategia, tratando de establecer cada posible resultado [Drucker, 2007].

Otro aspecto fundamental es el uso adecuado de los recursos para alcanzar el objetivo planteado, porque si se cuenta con una buena estrategia puede evitarse pérdida de tiempo [Contreras Sierra, 2013].

La estrategia relacionada con las actividades de gestión nos permite identificar cuál es nuestro proceso a realizar y la manera en que debe llevarse a cabo. Con la finalidad de cumplir con los objetivos o políticas planteados.

2.15. Segmentación

La segmentación es un esquema de asignación de memoria donde el espacio de direcciones lógicas se compone de un conjunto de segmentos, cada uno con un nombre y longitud [Fernández Redondo, 2006].

La segmentación de un recurso en tiempo de carga lo divide en tantos segmentos como sean necesarios, los cuales son almacenados en un repositorio de manera ordenada y no puede alterarse o perderse su secuencia, sin embargo, esto representa una dependencia entre la aplicación web y el repositorio para poder realizar su presentación.

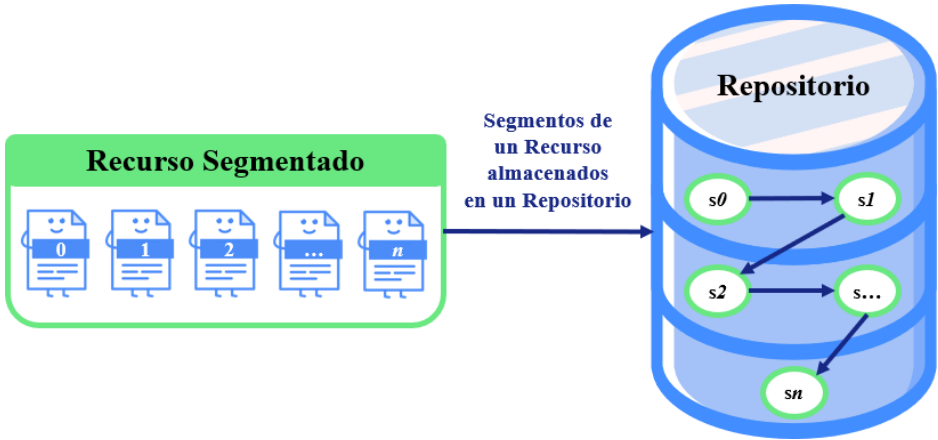


Figura 2.8. Segmentación en tiempo de carga

Por otro lado, la segmentación puede realizarse en tiempo de ejecución donde no es necesario el uso de repositorios, debido a que existe un desplazamiento automático para cada segmento generado lo que permite obtener su secuencia de manera ordenada. Como es el caso en este trabajo de tesis.

Los objetos segmentados pueden ser referenciados por medio de una dirección lógica de dos dimensiones: el número del segmento s y el desplazamiento de un segmento a otro d . Los números de los segmentos se utilizan como índice en una tabla de segmentos. Cada entrada de la tabla, tiene una base que corresponde al segmento inicial 0 y un límite que corresponde al segmento final n . El desplazamiento d de la secuencia tiene que estar comprendido entre 0 y n . En caso contrario se produce una interrupción que no permitiría tener la secuencia correcta de los segmentos. Si el desplazamiento es correcto, se suman todos los segmentos ordenadamente a la base para producir una dirección física perteneciente a la memoria virtual. La tabla de segmentos es esencialmente una matriz de registros base/límite (segmentos que van de 0 a n).

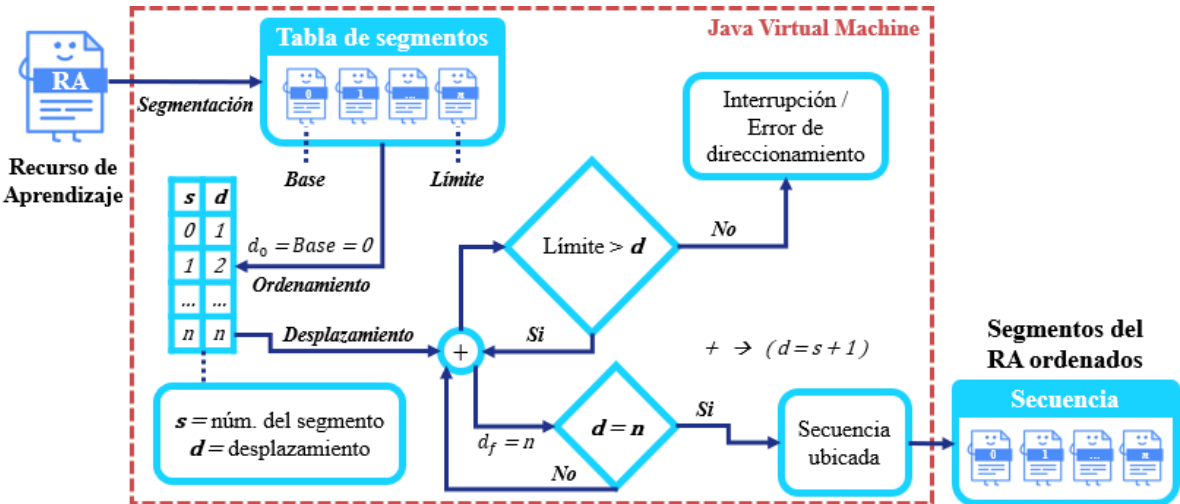


Figura 2.9. Segmentación en tiempo de ejecución

La segmentación de un recurso consiste en la división del mismo, en tantos segmentos como sean necesarios de acuerdo a un límite de tamaño asignado.

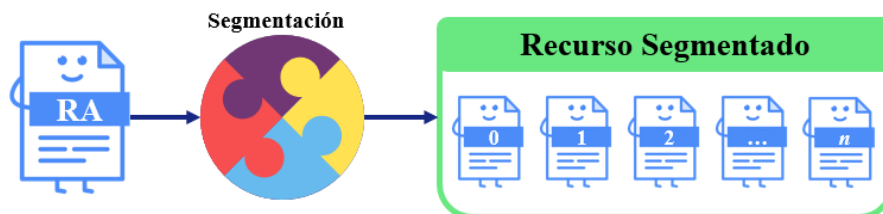


Figura 2.10. Segmentación de un recurso

2.16. Reconstrucción

La reconstrucción es el proceso de unir distintas partes que conforman a un objeto, se utiliza el número del segmento como un índice para tener un orden correcto y relacionar a cada uno con la parte que contiene. Al reconstruir se debe disponer de espacio suficiente de memoria para su generación [Navarro Luna, 2001].

La reconstrucción de un recurso consiste en la unión de múltiples segmentos ordenados y que conforman a un mismo producto para ser entregado.

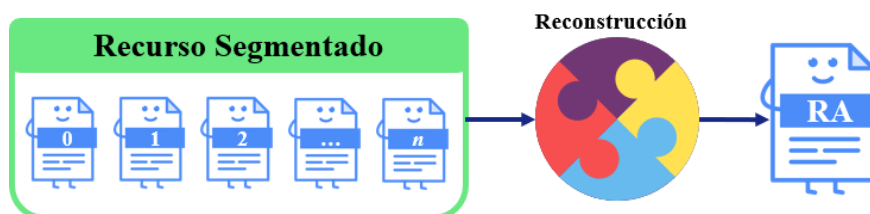


Figura 2.11. Reconstrucción de un recurso segmentado

2.17. MIME: Extensiones Multipropósito de Correo de Internet

El tipo Extensiones Multipropósito de Correo de Internet (MIME)¹ es una forma estandarizada de indicar la naturaleza y el formato de un documento. Está definido y estandarizado como IETF RFC 6838. La Autoridad de Números Asignados de Internet (IANA) es el organismo oficial responsable de realizar un seguimiento de todos los tipos MIME reconocidos [MDN Web Docs, 2019].

Los navegadores web a menudo usan el tipo MIME (y no la extensión o formato del recurso) para determinar cómo procesará un documento; por lo tanto, es importante que los servidores estén configurados correctamente para adjuntar el tipo MIME como encabezado del objeto de respuesta para poder visualizar a los recursos.

¹ La lista completa de Tipos de Medios (Media Types), conocidos como tipos MIME, se encuentra enlistada y en constante actualización por la IANA en <https://www.iana.org/assignments/media-types/media-types.xhtml>

Capítulo

3

Antecedentes & Trabajos Relacionados

En este capítulo se describen los antecedentes que dan soporte a este trabajo y se presentan los trabajos relacionados que más se asemejan a este contexto, los cuales se enfocan a las estrategias de gestión para el manejo de recursos o datos a través de servicios web.

3.1. Antecedentes de la Investigación

En el *Tecnológico Nacional de México (TecNM) / Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET)*, específicamente en el área de Ingeniería de Software, se han desarrollado diez tesis de maestría y una de doctorado donde se aborda la creación, manejo y control en la construcción recursos digitales con material educativo como servicios web, a los cuales se les denomina Servicios Web de Aprendizaje. Los antecedentes relacionados a este concepto se presentan en la Figura 3.1.

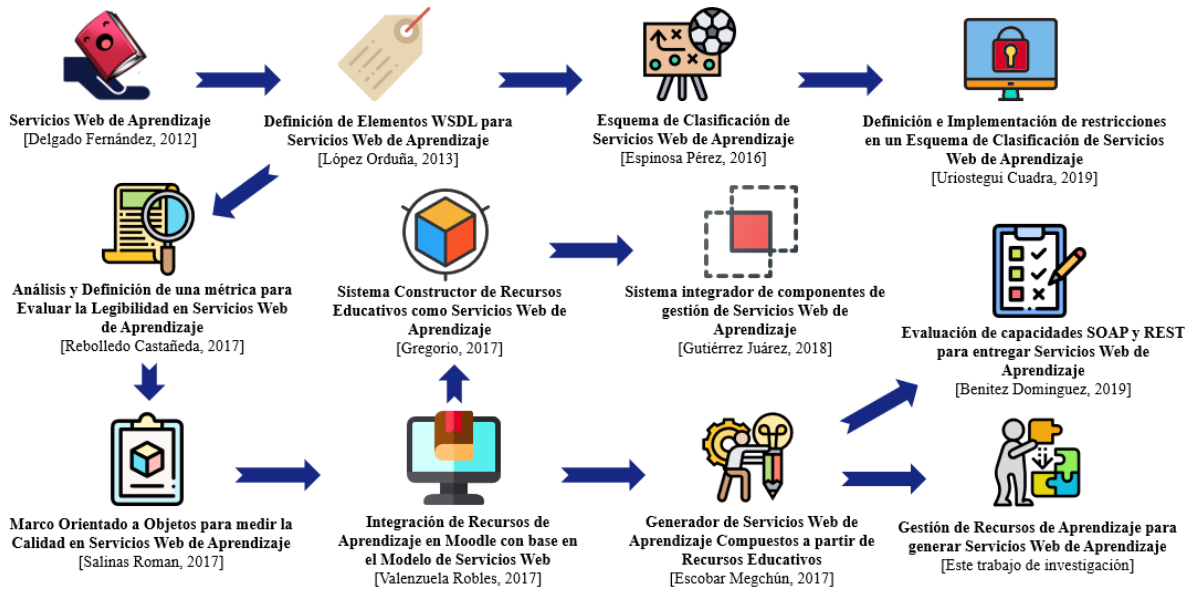


Figura 3.1. Antecedentes relacionados a los Servicios Web de Aprendizaje

A continuación, se describen cronológicamente los antecedentes de esta investigación:

Servicios Web de Aprendizaje [Delgado Fernández, 2012]. Este primer trabajo determina la viabilidad de construir recursos de aprendizaje en términos de servicios web, lo que llevó a proponer el término Servicio Web de Aprendizaje, definido en primera instancia como *la unidad de aprendizaje construida como servicio web que contiene el mecanismo de comunicación hacia el contenido educativo de uno de los cuatro elementos que conforman a un objeto de aprendizaje, funcionando como intermediario entre el usuario y el contenido de aprendizaje*. En este contexto, se trata a uno de estos cuatro elementos que conforman a los objetos de aprendizaje para empaquetarlo de manera individual como un servicio web, lo que permite la construcción de un servicio web con contenido educativo de tipo objetivo, contenido, actividad o evaluación. Este modelo proporciona un mayor alcance para distribuir el contenido educativo a través de la web al reducir las limitantes en el uso y facilitando la creación, actualización y disponibilidad de los recursos de aprendizaje.

Definición de Elementos WSDL para Servicios Web de Aprendizaje [López Orduña, 2013]. Este segundo trabajo identifica las instrucciones del archivo descriptor de los servicios web (WSDL) que deben modificarse para que los objetos de aprendizaje puedan ser manipulados a través de servicios web, y así desplegar los contenidos educativos que son

ejecutados. Se implementa una infraestructura para extender al esquema de clasificación del directorio jUDDI para integrar a los servicios web de aprendizaje dentro de una arquitectura orientada a servicios (SOA) para realizar los registros, búsquedas, localizaciones e invocaciones con mayor facilidad. Lo que permite su correcta implementación y gestión entre los distintos sistemas gestores de aprendizaje (SGA).

Esquema de Clasificación de Servicios Web de Aprendizaje [Espinosa Pérez, 2016]. Este tercer trabajo define e implementa un esquema de clasificación por dominio de aplicación para recuperar los elementos que describen a los servicios web de aprendizaje y mejor cumplan con la especificación de requerimientos de los usuarios, debido a que la UDDI, aunque sea un directorio de servicios web, no posee forma de clasificar que permita recuperar y seleccionar a los servicios de manera precisa. El esquema presenta un conjunto de descriptores representados en un árbol jerárquico semántico, donde se clasifican en cinco categorías: dominio o área de conocimiento, nivel escolar, formato de los recursos educativos, tipo de servicio web de aprendizaje, modalidades de aprendizaje y el idioma. Las cuales, forman subárboles con descriptores específicos de cada una y se basan en algoritmos de similitud para la recuperación de los servicios web de aprendizaje a partir de los requerimientos del usuario, obteniendo un alto índice de precisión y recuperación en la búsqueda y localización de los servicios requeridos.

Análisis y Definición de una métrica para Evaluar la Legibilidad en Servicios Web de Aprendizaje [Rebolledo Castañeda, 2017]. Este cuarto trabajo identifica y analiza atributos de calidad para evaluar el contenido que poseen los servicios web de aprendizaje, para apoyar su caracterización en sistemas de clasificación, selección y recuperación de servicios web de aprendizaje con el propósito de extender su reusabilidad. Así mismo, se plantean métricas y umbrales para los criterios tomados en cuenta de la evaluación propuestos para medir de manera manual la legibilidad de los servicios web de aprendizaje, dependiendo del tipo de contenido para documentos de texto o por diapositivas.

Marco Orientado a Objetos para medir la Calidad en Servicios Web de Aprendizaje [Salinas Roman, 2017]. Este quinto trabajo presenta un marco donde se describen atributos de calidad para evaluar el contenido en los servicios web de aprendizaje, cuya característica principal en este modelo de calidad es la reusabilidad, conformada por los sub-atributos: disponibilidad, usabilidad, accesibilidad e informatividad. Presenta un componente de software, que posee una arquitectura de marco orientado a objetos, la cual considera a los criterios tomados en cuenta para evaluar la calidad de forma objetiva y automática, con el propósito de contribuir en actividades de clasificación, selección y utilización de servicios web de aprendizaje.

Sistema Constructor de Recursos Educativos como Servicios Web de Aprendizaje [Gregorio, 2017]. Este sexto trabajo presenta una herramienta de autor para la construcción de recursos educativos y la generación automática de servicios web de aprendizaje. Los recursos pueden ser construidos desde cero, así como también de contenido educativo ya existente, y son generados en formato HTML. Los servicios web de aprendizaje generados con esta herramienta hacen uso de los protocolos HTTP, SOAP, WSDL y XML

como la tecnología base para la comunicación entre distintas plataformas, y los recursos educativos construidos son categorizados en los cuatro elementos pedagógicos correspondientes a los objetos de aprendizaje, es decir, la herramienta permite construir servicios web de aprendizaje de tipo: objetivo, contenido, actividad y evaluación.

Integración de Recursos de Aprendizaje en Moodle con base en Modelo de Servicios Web [Valenzuela Robles, 2017]. Este séptimo trabajo describe a los recursos de aprendizaje en términos de servicios web, denominados como servicios web de aprendizaje, que son componentes de baja granularidad y, por lo tanto, susceptibles de ser combinados en secuencias lógicas educativas. Cuyas secuencias deben estar encaminadas hacia la composición de un tema o unidad de aprendizaje. En este contexto, se muestran las ventajas que brindan los servicios web de aprendizaje como alternativa a los objetos de aprendizaje, mediante la extensión de un sistema gestor de aprendizaje (SGA), Moodle en el caso de este trabajo, donde se realiza la agregación, presentación y gestión de los servicios web de aprendizaje, con el propósito de mostrar la reusabilidad e independencia que tienen los recursos de aprendizaje a través de servicios web de aprendizaje.

Generador de Servicios Web de Aprendizaje Compuestos a partir de Recursos Educativos [Escobar Megchún, 2017]. Este octavo trabajo desarrolla una herramienta que permite la generación automática de servicios web de aprendizaje a partir de recursos de aprendizaje, los cuales pueden ser ingresados de manera local o referenciados desde un enlace de Internet, así también permite generar servicios compuestos que puedan invocar a múltiples recursos, con la finalidad de ser utilizados en un sistema gestor de aprendizaje (SGA) que soporte al aprendizaje electrónico basados en servicios web de aprendizaje. Los servicios generados pueden ser de tipo SOAP o REST, y contienen al recurso de manera codificada para evitar la dependencia entre el servicio y el recurso. Esta herramienta es esencial en la presente investigación, ya que, es el antecedente principal donde surge la problemática en este trabajo de tesis que consiste en el desbordamiento de memoria al ejecutar servicios web de aprendizaje con recursos de aprendizaje de gran tamaño.

Sistema integrador de componentes de gestión de Servicios Web de Aprendizaje [Gutiérrez Juárez, 2018]. Este noveno trabajo desarrolla un sistema que permite la integración de componentes de gestión de servicios web de aprendizaje, a través de una arquitectura extensible basada en servicios web para integrar tantos componentes como sean requeridos, con el propósito de ampliar su utilización en conjunto y su despliegue de manera independiente. El sistema presenta la integración de dos componentes, el primero es un modelo de calidad para medir la reusabilidad de los servicios web de aprendizaje y el segundo es un sistema evaluador de diversos dominios para calificar el aprendizaje en estudiantes.

Evaluación de capacidades SOAP y REST para entregar Servicios Web de Aprendizaje [Benítez Domínguez, 2018]. Este décimo trabajo evalúa la entrega de recursos de aprendizaje a través de los servicios web de aprendizaje de tipo SOAP y REST, considerando tres sub-atributos: tiempo de respuesta, *throughput* (emisión de información a través de un canal de comunicación en un rango de tiempo) y tasa de éxito. Presenta un modelo de calidad cuyo principal atributo es la entregabilidad, definida como *la capacidad*

de entregar un recurso de aprendizaje. De acuerdo a los resultados obtenidos mediante este modelo, la mejor alternativa para la entrega de recursos de aprendizaje es con la capacidad de los servicios web de aprendizaje basados en SOAP.

Definición e Implementación de restricciones en un Esquema de Clasificación de Servicios Web de Aprendizaje [Uriostegui Cuadra, 2019]. Este undécimo trabajo define e implementa un conjunto de restricciones para un esquema de clasificación de servicios web de aprendizaje, con el propósito de prevenir defectos al clasificar servicios derivados del uso incorrecto de palabras clave. Estas restricciones permiten recuperar con precisión los servicios web de aprendizaje que se requieren para cumplir con los requerimientos de un usuario, basándose en el análisis de catálogos educativos y la metodología casa de la calidad.

3.2. Trabajos Relacionados

El estudio de trabajos relacionados permitió identificar dos enfoques sobre las distintas “estrategias de gestión” y acerca de la “generación y composición de servicios web”.

3.2.1. Trabajos de Estrategias de Gestión

En esta sección se detallan trabajos relacionados que se enfocan a distintas estrategias de gestión como son los casos de la segmentación, reconstrucción, clasificación, compactación, verificación, entre otras estrategias para el control y manejo de los recursos o del flujo de datos.

Data Stream Management [Golab & Ozsu, 2010]. Este trabajo aborda la temática que tienen las aplicaciones para el procesamiento de grandes tamaños en datos de transmisión, como es el caso del tráfico de Internet. Un flujo de datos (data flow) es un conjunto de datos ilimitados que se produce de manera incremental a lo largo del tiempo, en lugar de estar disponible en su totalidad antes de que comience su procesamiento. El procesamiento de flujos de datos implica desde responder consultas sencillas en flujos de alta velocidad hasta la carga de feeds de datos en tiempo real en un almacén de flujos para realizar análisis fuera de línea. Donde se comparan dos tipos de sistemas para el procesamiento de flujo de extremo a extremo: Data Stream Management Systems (DSMS) y Streaming DataWarehouses (SDW). El trabajo se enfoca en la gestión de flujo de datos en DSMS para el procesamiento de consultas en línea y en SDW para el análisis fuera de línea. A medida que aumenta la cantidad de datos de transmisión, los DSMS deben volverse aún más escalables que en la actualidad, una solución es aprovechar el nuevo hardware. Mientras que los SDW requieren actualización de las estrategias de programación para optimizar diversos objetivos, y monitorear la consistencia y la calidad de los datos a medida que llegan nuevos datos.

A Unified Record Linkage Strategy for Web Service Data [Kan, Yang, Zhen, & Liu, 2010]. Este trabajo propone un marco para reconocer grupos de registros duplicados aproximados de datos en varios idiomas funcionando en tres pasos. La vinculación de

registros, o detección de duplicados, es un proceso clave que garantiza la calidad de los datos almacenados para el contenido del servicio web. Dadas dos listas de registros, la vinculación de registros consiste en determinar todos los pares que son similares entre sí, donde la similitud general entre dos registros se define en función de las similitudes específicas de dominio sobre los atributos individuales que constituyen el registro. La estrategia de gestión consiste en el procesamiento previo de datos en varios idiomas utilizando segmentación de palabras y técnicas de reconocimiento de entidades con nombre en chino; un método de comparación por pares basado en similitudes de dominio específico del kernel de cadenas para determinar similitudes específicas de dominio entre atributos multilingües; una cola de prioridad de clústeres duplicados y una estrategia de registros representativos para responder de manera adaptativa a la escala de datos. Los resultados experimentales realizados en bases de datos reales, muestran que la estrategia de enlace de recodificación tiende hacia la eficiencia y la escalabilidad.

Automatic management of cyclic dependency among web services [Omer & Schill, 2011]. Este trabajo propone un método para identificar, extraer y manejar automáticamente las dependencias cíclicas entre servicios, surgidos de la conexión entre los componentes de servicios y las restricciones en su interacción. Una dependencia cíclica es uno de los diferentes tipos de dependencias potenciales que podrían ocurrir entre los servicios web. El enfoque propuesto considera dos formas de representar la dependencia, mediante matriz y gráfico dirigido, donde se proporcionan formas de lidiar con la dependencia cíclica para ambos casos. Se presenta el algoritmo que permite el ordenamiento y la ejecución entre servicios web con dependencia directa mediante ciclos para el caso de matrices y por nodos compuestos para el caso de gráficos dirigidos. La estrategia de gestión expuesta permite identificar y extraer la dependencia cíclica de la matriz o gráfica de dependencia del servicio. De igual modo proporciona una forma para regenerar el gráfico de dependencia acíclica o la matriz para cualquier aplicación que utilice la dependencia del servicio.

Adaptative CDCLZW algorithm for data compression [Liu, Liao, & Shen, 2013]. Este trabajo propone un algoritmo de compresión basado en Content Defined Chunking (CDC) y Lempel-Ziv-Welch (LZW) para comprimir los datos JSON en la medida de lo posible y ahorrar ancho de banda entre los complementos del cliente y los servidores de transferencia central, llamado Content Defined Chunking Lempel Ziv Welch (CDCLZW), cuyo propósito es el rastreo de flujo de código para servidores web. El algoritmo CDCLZW utiliza una Librería de Plantillas y el esquema de implementación polinomial de Rabin Fingerprint para encontrar datos duplicados y evitar su transferencia entre clientes y servidores. El proceso de compresión se enfoca en un algoritmo basado en CDC para realizar la segmentación por tamaño y parámetros de valor que eliminen partes del contenido, posteriormente se aplica un algoritmo basado en LZW para el proceso de descompresión que permita componer o recuperar el contenido segmentado. Se realizaron pruebas que demuestran que CDCLZW puede asegurar la relación de compresión constante y obtener mejores resultados que los algoritmos GNUzip (gzip) y bzip2.

e-Learning Management System Using Web Services [Partheeban & SankarRam, 2014]. Este trabajo define un sistema arquitectónico de e-learning en tres niveles con el

objetivo de proporcionar una base a los diseñadores, desarrolladores e instructores que permitan construir modelos estratégicos de e-learning para sus entornos. El marco propuesto utiliza el enfoque de servicios web para aumentar su eficiencia y la efectividad del aprendizaje colectivo en términos de reutilización, interoperabilidad, accesibilidad y modularización. El método de creación propuesto indica que los servicios web brindan una mejor alternativa al presentar bajo acoplamiento y aprovechar los estándares de entornos web en el e-learning, facilitan la integración dinámica entre sistemas distribuidos, al transformar archivos HTML a XML, y fomentan la reutilización de los objetos de aprendizaje.

Probabilistic chunk scheduling approach in parallel multiple-server DASH [Liu, Zhou, Zhang, Guo, & Li, 2014]. Este trabajo propone una estrategia para la descarga consecutiva de segmentos con Dynamic Adaptive Streaming HTTP (DASH), la tecnología DASH tiene el propósito de proporcionar un mayor ancho de banda, diversidad de conexiones y confiabilidad. La estrategia se enfoca a un estilo de programación probabilístico de segmentación considerando el ancho de banda variable en el tiempo, los segmentos de vídeo se programan en servidores que consumen el menor tiempo y presentan una mayor probabilidad de completar la descarga antes del tiempo límite. Este enfoque minimiza el tiempo total de la descarga del vídeo, primero realiza la estimación probabilística de descargar los segmentos exitosamente antes de concluir con el tiempo límite de reproducción del recurso multimedia. Luego se estima el tiempo de descarga de los segmentos, de manera que se optimice el menor tiempo de descarga posible. Los resultados experimentales indican que aumentan en mayor cantidad el número de segmentos que se reciben de manera ordenada, a diferencia de otros esquemas similares.

Two-Stream Deep Encoder-Decoder Architecture for Fully Automatic Video Object Segmentation [Xu, Song, & Xie, 2017]. Este trabajo propone una arquitectura para profundizar la codificación-decodificación en dos secuencias mediante flujos de convolución de manera automática en objetos de vídeo. La primera secuencia consiste en la segmentación estática de la imagen y la segunda en la segmentación del flujo óptico. El codificador genera una máscara de baja resolución con ubicaciones precisas y límites suaves, mientras que el decodificador refina los detalles de la máscara inicial y aumenta su resolución al integrar progresivamente las funciones de nivel inferior. Estos dos flujos aprenden a integrarse para obtener mejores resultados. Además, para manejar el problema de los conjuntos de datos de segmentación de objetos de video inadecuados, se propone una estrategia de búsqueda para generar un conjunto de datos a gran escala para su capacitación. Los experimentos en dos conjuntos de datos estándar demuestran que el método propuesto supera a la mayoría de los métodos de vanguardia en precisión de segmentación y tiempo de ejecución.

Faster Base64 Encoding and Decoding Using AVX2 Instruction [Muła & Lemire, 2018]. Este trabajo propone una estrategia de segmentación mediante instrucciones de algoritmos de extensiones vectoriales avanzadas (AVX2) que permiten aumentar la velocidad en la codificación hacia una proximidad de 10 veces mayor y en la decodificación hacia una proximidad de 7 veces mayor de datos base64. La codificación y decodificación de datos base64 implica la asignación de bloques de 3 bytes a bloques de 4 caracteres ASCII. Normalmente se decodifican los datos accediendo a cada carácter, uno por uno. Para una

mejor velocidad, es posible optar por decodificar datos utilizando registros de vectores completos. Dependiendo de la mayor longitud de vector disponible, podemos decodificar datos base64 en conjuntos de 16, 32 o incluso 64 caracteres a la vez. Este enfoque asume que los datos originales codificados en base64 son divisibles por 24 bytes. Sin embargo, se procesan los datos restantes usando uno de los algoritmos escalares propuesto en este trabajo.

A Strategy of Visualization and Interactive Support for University Level Educational Digitalization [Kolesnikov, Lomachenko, Kokodey, Khitushchenko, & Mihailov, 2019]. Este trabajo implementa una estrategia de visualización y soporte interactivo para la digitalización educativa a nivel universitario. Desarrolla un modelo que visualiza el proceso educativo a través de un sistema gestor de aprendizaje (SGA) constituido por dos módulos: servicios web externos y actividades interactivas de Moodle. Los servicios web brindan la visualización del material educativo mediante el uso de distintos dominios educativos, mientras que las actividades de Moodle se utilizan para realizar pruebas de conocimiento, como actividades y evaluaciones, y permite la comunicación entre los estudiantes, a través de foros y seminarios. El modelo propuesto utiliza cursos de e-learning propios, desarrollados por la Universidad Estatal de Sevastopol (SevSU) en Moodle.

3.2.2. Trabajos de Generación y Composición de Servicios Web

En esta sección se detallan trabajos relacionados que se enfocan a la generación y la composición de servicios web con contenidos educativos o que consideren ciertos criterios del aprendizaje electrónico o implementen estrategias de gestión mediante servicios web.

Method of creating web services from web applications [Nakano, Yamato, Takemoto, & Sunaga, 2007]. Este trabajo propone un método que permite utilizar mediante wrappers (envoltorios) a las aplicaciones web como servicios web. El proceso consiste en analizar los documentos HTML de la aplicación y segmentarlos de acuerdo a sus etiquetas, donde se establece una serie de reglas para su extracción mediante envolturas, utilizando la profundidad de cada documento HTML. El método cuenta con una estrategia que realiza seis actividades donde se involucran procesos de búsqueda, segmentación, clasificación y localización de etiquetas HTML. Con la generación y el uso de las envolturas es posible crear componentes de servicios web y servicios compuestos que utilizan esos componentes por sí mismos, los resultados experimentales indicaron que el método propuesto logra extraer exitosamente un 37% de los documentos HTML generados por las aplicaciones web.

Symbolic model checking composite Web services using operational and control behaviors [Bentahar, Yahyaoui, Kova, & Maamar, 2013]. Este trabajo presenta un modelo para servicios web compuestos en función de una separación entre aspectos de negocios y de control de los servicios web. Indaga una problemática desde el diseño de los servicios web compuestos para verificar si cumplen con ciertas propiedades deseables en términos de libertad de interbloqueo (deadlock freedom), seguridad y accesibilidad (reachability). La separación se realiza a través del diseño de un comportamiento operativo, que define la composición de acuerdo con la lógica de negocios de los servicios web, y un comportamiento

de control, que identifica las secuencias válidas de acciones que debe seguir el comportamiento operativo. Estos dos comportamientos se definen formalmente utilizando técnicas basadas en automatización. El enfoque propuesto se basa en la verificación de modelos, donde el comportamiento operacional es el modelo que debe verificarse con las propiedades definidas en el comportamiento de control. El artículo demuestra que la técnica propuesta permite verificar la solidez y la integridad del modelo de diseño con respecto a los comportamientos operativos y de control.

Ontology based dynamic e-learning flow composition of Learning web services [Begam & Ganapathy, 2014]. Este trabajo sugiere una arquitectura de composición basada en el flujo de trabajo para los servicios web de aprendizaje y algoritmos relevantes, con el propósito de componer secuencias de e-learning para distintos tipos de aprendizaje. El trabajo recomienda el uso de algoritmos de combinación y la composición no-híbrida basada en la lógica que usan los perfiles OWL-S y procesos de ontología para la composición dinámica de las secuencias de servicios web de aprendizaje. En este artículo se considera a un servicio web de aprendizaje a cualquier material educativo encapsulado como servicio web, pero no necesariamente clasificados como alguno de los tipos de objetos de aprendizaje: objetivo, contenido, actividad y evaluación.

Creating e-Learning Web Services Towards Reusability of functionalities in creating e-learning systems [Rabahallah & Ahmed-Ouamer, 2015]. Este trabajo crea un conjunto de servicios web de aprendizaje que permiten construir nuevos sistemas gestores de aprendizaje (SGA) en base a la reutilización de funcionalidades. Donde se identifican a cuatro funcionalidades principales: construcción de capítulos, generación de ejercicios, generación de evaluaciones y generación de simuladores. Las cuales fueron construidas como componentes reusables en términos de servicios web, haciendo uso de la ontología OWL-S para describirlas como servicios web de aprendizaje.

Set partition and trace based verification of Web service composition [Rai & Gangadharan, 2015]. Este trabajo presenta un modelo para el análisis y verificación en los procesos de composición de los servicios web a nivel de diseño. Donde se particionan a los servicios en varios subconjuntos en función del orden de su invocación, se organizan para formar un gráfico de partición del conjunto de servicios web y transformarlos en un conjunto de trazas interactivas. Se propone una metodología para la verificación de la interacción del servicio que utiliza la descripción del servicio (WSDL) para extraer la información necesaria y facilitar el proceso de modelado, análisis y razonamiento de los servicios compuestos. Esta técnica de partición hace que un conjunto de servicios web se consideren candidatos para un escenario de composición en una serie de subconjuntos basados en la posibilidad de invocación del servicio y, por lo tanto, generen un gráfico de partición de conjunto de servicios web (WSSP). Como se aprecia en la figura, donde se tiene una vista consolidada de un escenario de composición de servicios web, a la cual se realiza una vista distinguida del subconjunto de servicios web que son posibles candidatos. Se presentan dos definiciones formales de terminologías relacionadas con el rastreo para modelar y de razonar las interacciones de una composición de servicio web.

Effective service composition using multi-agent reinforcement learning [Wang, Wang, Zhang, Yu, & Hu, 2015]. Este trabajo propone un marco para la composición dinámica de servicios web complejos, basados en el reforzamiento de agentes múltiples para que aprendan a adaptarse en entornos dinámicos. Todos los agentes implementados trabajan en conjunto para realizar una tarea común, utilizando una estrategia de búsqueda distribuida, cada agente lleva a cabo un proceso de aprendizaje considerando sub-objetivos que permiten acelerar la eficiencia del sistema. El trabajo realiza un experimento para comparar al marco propuesto con distintos aspectos y técnicas de aprendizaje, donde se concluye que el algoritmo tiene la capacidad de adaptación y una mayor eficiencia para la composición dinámica de servicios.

Multi-Agent Based Model for Web Service Composition [Belmabrouk, Bendella, & Bouzid, 2016]. Este trabajo propone un modelo basado en agentes múltiples para la planificación automática de servicios web y garantizar la calidad requerida para un servicio web compuesto. El marco se encuentra distribuido por un conjunto de diversos agentes que completan sus tareas para alcanzar el objetivo común planteado por el usuario, dichos agentes crean un plan global que permita definir la composición de los servicios web.

Adaptative composition in dynamic service environments [Barakat, Miles, & Luck, 2018]. Este trabajo desarrolla un algoritmo de ejecución adaptativa como estrategia para la composición de servicios web, capaz de gestionar cambios en el servicio en tiempo de ejecución para fines de reparación y optimización. La adaptación se realiza en paralelo con el proceso de ejecución, reduciendo los tiempos de interrupción y aumentando las posibilidades de recuperación exitosa. El algoritmo identifica con base en la categorización de los cambios, especificando su urgencia e importancia, y guiando el comportamiento del sistema en ejecución. Cuando un servicio ejecutado entrega valores de calidad imprevistos, el algoritmo logra recuperarse de la situación con una interrupción mínima.

3.3. Resumen Comparativo de los Trabajos Relacionados

3.3.1. Comparación de los Trabajos de Estrategias de Gestión

En la Tabla 3.1 se presenta un resumen comparativo de los trabajos relacionados al trabajo de tesis enfocados a distintas estrategias de gestión, en la cual se utilizan los siguientes criterios de comparación:

- **Contexto:** se refiere a lo que se presume que se hizo en cada uno de los trabajos relacionados, se distinguen tres casos: compresión, detección y gestión.
- **Enfoque:** se refiere al enfoque trabajado en cada uno de los trabajos analizados, se distinguen tres casos: flujo de datos (FD), servicio web (SW) y servicio web de aprendizaje (SWA).
- **Objetivo:** se describe brevemente el propósito o lo que se quiere lograr mediante de la realización del trabajo.
- **Recursos:** este criterio se utiliza para indicar los recursos utilizados en cada uno de los trabajos analizados, se distinguen cuatro casos: datos (D), recursos de aprendizaje (RA), recursos web (RW) y recursos multimedia (RM).
- **Estrategia:** es la estrategia implementada que realizo cada uno de los trabajos analizados.
- **Producto o Resultado:** representa el producto final obtenido en cada uno de los trabajos analizados.

Tabla 3.1. Resumen comparativo de trabajos relacionados a estrategias de gestión

Cita del Trabajo	Contexto	Enfoque	Objetivo	Recursos	Estrategia	Producto o Resultado
[Golab & Ozsu, 2010]	Gestión	FD	Procesar grandes tamaños de datos de transmisión	RW	Una estrategia de gestión de flujo de datos para el procesamiento de consultas en línea, y una segunda estrategia para el análisis fuera de línea en la cantidad de datos de transmisión	Dos arquitecturas que permiten gestionar las cadenas de datos de acuerdo a sus contenidos
[Kan, Yang, Zhen, & Liu, 2010]	Detección	SW	Garantizar la calidad de los datos almacenados en el contenido de servicios web	D	Una estrategia de vinculación de datos para la detección de datos duplicados a través de la segmentación de palabras	Un algoritmo de enlace de recodificación que tiende a la eficiencia y escalabilidad en la reducción de datos duplicados
[Omer & Schill, 2011]	Gestión	SW	Obtener la dependencia cíclica de la matriz o gráfica de dependencia del servicio	RW	Una estrategia de identificación, extracción y gestión automática de las dependencias cíclicas en servicios web	Un método para la gestión de dependencias cíclicas surgidas de la conexión entre componentes de servicios y restricciones de interacción

[Liu, Liao, & Shen, 2013]	Compresión	FD	Rastrear el flujo de datos entre servidores web	D	Una estrategia de compresión para la reducción en el tamaño de las cadenas de datos	Un algoritmo para comprimir cadenas de datos JSON que ahorren ancho de banda entre clientes y servidores
[Partheeban & SankarRam, 2014]	Gestión	SWA	Construir modelos estratégicos de aprendizaje electrónico para sistemas gestores de aprendizaje	RA	Una estrategia de creación de servicios web de aprendizaje para fomentar la reutilización de los objetos de aprendizaje entre plataformas educativas	Un marco en tres niveles que permite construir modelos estratégicos de aprendizaje electrónico para aumentar la eficiencia y efectividad del aprendizaje en términos de reutilización, interoperabilidad, accesibilidad y modularización
[Liu, Zhou, Zhang, Guo, & Li, 2014]	Gestión	SW	Presentar vídeos de gran tamaño en el menor tiempo de respuesta posible	RM	Una estrategia de estilo de programación probabilístico de segmentación considerando el ancho de banda variable en el tiempo	Un método para la descarga consecutiva de segmentos con uso de la tecnología Dynamic Adaptive Streaming HTTP (DASH)
[Xu, Song, & Xie, 2017]	Gestión	FD	Aumentar la precisión de segmentación y el tiempo de ejecución de vídeos	RM	Una estrategia de segmentación estática de la imagen y del flujo óptico en objetos de vídeo	Una arquitectura para profundizar la codificación-decodificación en dos secuencias mediante flujos de convolución de manera automática en objetos de vídeo
[Muła & Lemire, 2018]	Gestión	FD	Incrementar la velocidad en procesos de codificación-decodificación de datos base64	RA	Una estrategia de segmentación mediante instrucciones de algoritmos de extensiones vectoriales avanzadas (AVX2)	Un método de segmentación AVX2 que permite aumentar la velocidad de codificación hacia una proximidad de 10 veces mayor y de decodificación hacia una proximidad de 7 veces mayor de datos base64
[Kolesnikov, Lomachenko, Kokodey, Khitushchenko, & Mihailov, 2019]	Gestión	SWA	Presentar e interactuar contenido educativo a través de servicios web de aprendizaje	RA	Una estrategia de visualización y soporte interactivo en el proceso educativo, a través de una estructura entre servicios web y plataformas educativas, como caso de estudio a Moodle	Un modelo que visualiza el proceso educativo a través de un sistema gestor de aprendizaje constituido por dos módulos: servicios web externos y actividades interactivas de Moodle

Tesis: “Gestión de RA para generar SWA”	Gestión	SWA	Presentar servicios web de aprendizaje con contenidos de gran tamaño	RA	Una estrategia de segmentación y reconstrucción de recursos de aprendizaje encapsulados como servicio web	Un método para la gestión de recursos de aprendizaje que permita generar y presentar servicios web de aprendizaje con contenidos de gran tamaño
---	---------	-----	--	----	---	---

3.3.2. Comparación de los Trabajos de Generación y Composición de Servicios Web

En la Tabla 3.2 se presenta un resumen comparativo de los trabajos relacionados al trabajo de tesis enfocados a la generación y composición de servicios web, en la cual se utilizan los siguientes criterios de comparación:

- **Contexto:** se refiere a lo que se presume que se hizo en cada uno de los trabajos relacionados, se distinguen dos casos: composición y generación.
- **Enfoque:** se refiere al enfoque trabajado en cada uno de los trabajos analizados, se distinguen dos casos: servicio web (SW) y servicio web de aprendizaje (SWA).
- **Objetivo:** se describe brevemente el propósito o lo que se quiere lograr mediante de la realización del trabajo.
- **Recursos:** este criterio se utiliza para indicar los recursos utilizados en cada uno de los trabajos analizados, se distinguen cuatro casos: datos (D), recursos de aprendizaje (RA), recursos web (RW), recursos multimedia (RM).
- **Método:** es el método implementado que realizó cada uno de los trabajos analizados.
- **Producto o Resultado:** representa el producto final obtenido en cada uno de los trabajos analizados.

Tabla 3.2. Resumen comparativo de trabajos relacionados a generación y composición de servicios web

Cita del Trabajo	Contexto	Enfoque	Objetivo	Recursos	Método	Producto o Resultado
[Nakano, Yamato, Takemoto, & Sunaga, 2007]	Generación	SW	Crear servicios web a partir de aplicaciones web	RW	Un método de búsqueda, segmentación, clasificación y localización de etiquetas HTML que permite utilizar aplicaciones web como servicios web independientes	Un algoritmo de profundización que segmenta la frecuencia de etiquetas HTML mediante envoltorios
[Bentahar, Yahyaoui, Kova, & Maamar, 2013]	Composición	SW	Separar aspectos de negocios y de control de servicios	RW	Un método de verificación de las propiedades deseables, en términos de libertad de interbloqueo, para componer servicios web	Un modelo de composición de servicios web para separar aspectos de negocios y aspectos de control de los servicios

[Begam & Ganapathy, 2014]	Composición	SWA	Componer secuencias de aprendizaje electrónico para distintos tipos de aprendizaje	RA	Un método basado en procesos de ontología para la composición dinámica de las secuencias de servicios web de aprendizaje	Una arquitectura de composición basada en el flujo de trabajo de los servicios web de aprendizaje y algoritmos de combinación y composición no-híbrida relacionados a procesos de ontología dinámica
[Rabahallah & Ahmed-Ouamer, 2015]	Generación	SWA	Construir sistemas gestores de aprendizaje a partir de un conjunto de servicios web	RA	Un método basado en la ontología OWL-S, que describe como servicios web de aprendizaje a componentes reusables de servicios web que, construyen capítulos, generan ejercicios, evaluaciones y simuladores	Un conjunto de servicios web de aprendizaje que permite construir nuevos sistemas gestores de aprendizaje en base a la reutilización de funcionalidades
[Rai & Gangadharan, 2015]	Composición	SW	Analizar y verificar procesos de composición de servicios web a nivel de diseño	RW	Un método de verificación de la interacción del servicio que utiliza el WSDL para extraer la información necesaria y facilitar el proceso de modelado, análisis y razonamiento de los servicios compuestos	Un método para la verificación y extracción de información en un servicio web para facilitar el modelado de los procesos de composición
[Wang, Wang, Zhang, Yu, & Hu, 2015]	Composición	SW	Acelerar la eficiencia del sistema en entornos dinámicos	RW	Un método de composición de servicios web con la capacidad de adaptarse en entornos dinámicos que presenta una mayor eficiencia comparada con distintos aspectos y técnicas de aprendizaje	Un marco para la composición dinámica de servicios web complejos, basados en el reforzamiento de agentes múltiples para que aprendan a adaptarse en entornos dinámicos
[Belmabrouk, Bendella, & Bouzid, 2016]	Composición	SW	Garantizar la calidad requerida de un servicio web compuesto	RW	Un método distribuido por un conjunto de diversos agentes que completan sus tareas para alcanzar el objetivo común planteado por el usuario	Un modelo multi agente para la planificación automática de servicios web que garantice la calidad requerida del servicio web compuesto
[Barakat, Miles, & Luck, 2018]	Composición	SW	Gestionar cambios en el servicio web en tiempo de ejecución para fines	RW	Un método de ejecución adaptativa para la composición de servicios web en paralelo con el proceso de ejecución	Un algoritmo que identifica con base en la categorización de los cambios, especificando su

			de reparación y optimización			urgencia e importancia, y guiando el comportamiento del sistema en ejecución, reduciendo los tiempos de interrupción y aumentando las posibilidades de recuperación exitosa
Tesis: “Gestión de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje”	Generación y composición	SWA	Generar y componer servicios web de aprendizaje con contenidos de gran tamaño	RA	Un método que permita generar servicios web con recursos segmentados para ser presentados sin que se vean afectados por el tamaño	Un método para generar servicios web de aprendizaje que contengan recursos de aprendizaje, en caso de ser un recurso de gran tamaño se realice una segmentación de éste, y otro método para componer a servicios con contenidos de distintos tamaños

El presente trabajo de tesis consiste en definir e implementar una estrategia de gestión de Recursos de Aprendizaje que se encuentran codificados y encapsulados en servicios web, que permita segmentarlos y reconstruirlos para que no se vean afectados por la limitante en el almacenamiento dinámico de la aplicación. En algunos de los trabajos revisados se trabaja principalmente con recursos multimedia como es el caso de música o vídeos que tienen como objetivo principal el entretenimiento y no el aprendizaje, o con recursos web que buscan facilitar la presentación de componentes o archivos HTML. Cabe mencionar que una tendencia actual muy demandada es la transmisión por secuencias *streaming*, que simula la reproducción de recursos multimedia en tiempo real. Sin embargo, este método es utilizado únicamente en formatos de tipo vídeo o audio, y no necesariamente se emplean para propósitos de aprendizaje. Mientras que los recursos de aprendizaje pueden tener formatos de animación, audio, imagen, texto, vídeo, presentación por diapositivas, web, entre muchos otros. Dentro de las principales diferencias de este trabajo de tesis contra los descritos anteriormente, se pueden mencionar:

- Se generan servicios web cuyos contenidos son recursos con material educativo codificados en Base64.
- Se realiza una estrategia de gestión a nivel de código donde se segmentan y reconstruyen recursos en tiempo de ejecución.
- Se comparan dos tipos de servicios web para la transferencia de datos, a través del protocolo SOAP y de la arquitectura REST.
- No hay restricción en el formato del recurso para generar el servicio web, pero solo podrá ser visualizado si es de tipo MIME.

Metodología de Solución

En este capítulo se describe la metodología de solución utilizada para lograr el objetivo en este trabajo de investigación, el cual consiste en definir e implementar una estrategia de gestión para Recursos de Aprendizaje de gran tamaño contenidos en Servicios Web de Aprendizaje, de manera que se establezcan las reglas necesarias para poder ejecutar la correcta presentación de estos servicios.

4.1. Metodología de Solución

La metodología de solución consiste en definir e implementar una estrategia de gestión de los Recursos de Aprendizaje, de manera que se realice su segmentación en caso de ser necesario y se establezcan las reglas para poder reconstruirlo y empaquetarlo como un Servicio Web de Aprendizaje, para poder presentarlo adecuadamente sin verse afectado por su tamaño.

La Figura 4.1 representa de manera general las actividades que se realizaron para atender el problema planteado en el [Capítulo 1](#).

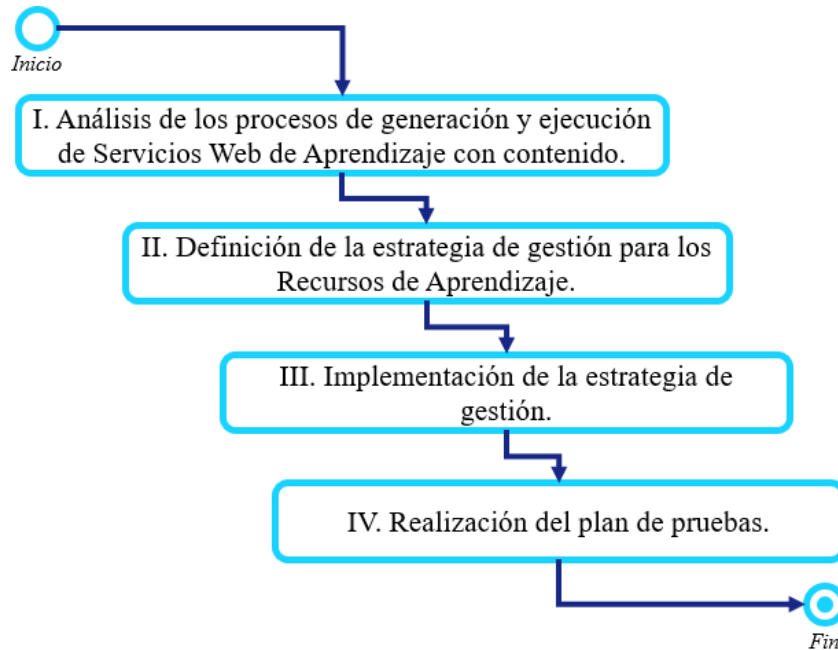


Figura 4.1. Metodología de solución

I. Análisis de los procesos de generación y ejecución de SWA con contenido

En la tesis [Escobar Megchún, 2017] se desarrolla una herramienta que permite la generación de Servicios Web de Aprendizaje a partir de recursos de aprendizaje, los cuales pueden ser ingresados de manera local o referenciados desde un enlace de Internet, así también para crear servicios compuestos que puedan invocar a múltiples recursos. Con la finalidad de ser utilizados en un Sistema Gestor de Aprendizaje que soporte al aprendizaje electrónico basados en SWA.

El proceso para la generación de un Servicio Web de Aprendizaje se observa en la Figura 4.2, donde tenemos a un recurso de aprendizaje como entrada el cual es codificado en Base64 y empaquetado como un servicio web, donde surge la problemática descrita en el [Capítulo 1](#), que impide presentar SWA mayores a los 50 MB. Ya que, al no contar con una estrategia de gestión, el recurso contenido se ve afectado por su infraestructura cuando es demasiado grande debido a que ocasiona que no puede ser presentado de manera correcta.



Figura 4.2. Proceso de generación de un SWA en la herramienta antecedente [Escobar Megchún, 2017]

Durante este proceso se decidió definir la estrategia de gestión, ya que al momento de codificar el recurso de aprendizaje a Base64 se genera una cadena demasiado larga, la cual puede ser analizada para aplicar una condición de segmentación a partir de un determinado tamaño.²

Esto a su vez, generaría un problema al contener el recurso segmentado en un servicio web debido a que no existiría un orden en el llamado hacia sus segmentos. Por lo que se planteó modificar al archivo WSDL en el caso de servicios SOAP, añadiendo una nueva etiqueta que referencia a cada segmento y el orden correcto para poder reconstruir al recurso de aprendizaje. Y en el caso de REST, se modifica a la clase con los métodos de implementación que conforman al servicio añadiendo un arreglo con los elementos necesarios para su reconstrucción.

II. Definición de la Estrategia de Gestión

Para el planteamiento de la estrategia se identificaron dos casos de uso a implementar como modificación por extensión a la herramienta antecedente, que consisten en la adición de dos actividades para la segmentación y la reconstrucción del recurso de aprendizaje, las cuales se observan en la Figura 4.3.

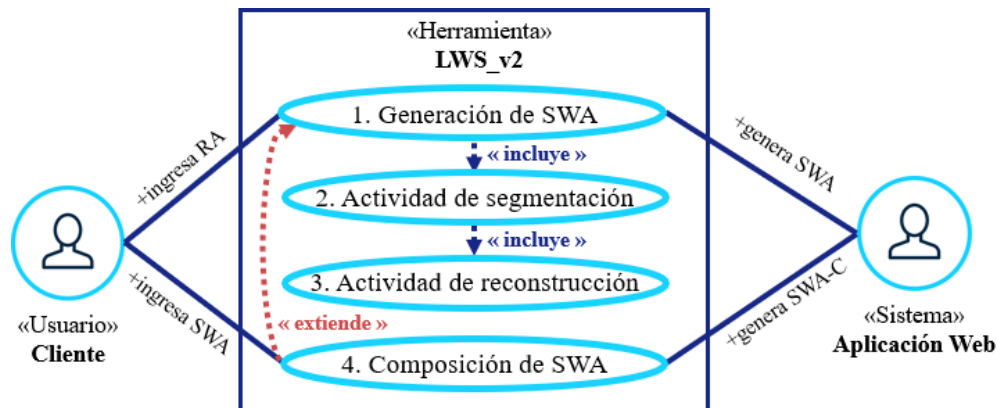


Figura 4.3. Diagrama de casos de uso

La Figura 4.3 representa los escenarios de casos de uso, donde el Caso de Uso 1 “Generación de SWA” permite al usuario ingresar un recurso de aprendizaje al sistema y generar un

² Un recurso debe ser codificado a Base64 para que pueda ser visualizado por un navegador web como recurso de tipo MIME.

Servicio Web de Aprendizaje. Al realizarse el Caso de Uso 1 se incluye el Caso de Uso 2 “Actividad de segmentación” solo sí se ingresa un recurso de gran tamaño, donde a partir de un determinado límite, se aplicará una condición para segmentar en tantas partes como sea necesario. Si la herramienta hace uso de esta actividad, entonces se incluirá también al Caso de Uso 3 “Actividad de reconstrucción” el cual modifica al documento WSDL del servicio web para poder ordenar los segmentos resultantes en el orden en que será reconstruido el recurso y que sea presentado. Independientemente que se aplique la segmentación o no, el Caso de Uso 4 “Composición de SWA” permite al usuario la generación de un Servicio Web de Aprendizaje Compuesto a partir de la entrada de múltiples Servicios Web de Aprendizaje.

Finalmente, la estrategia de gestión se lleva a cabo durante el proceso de generación de un Servicio Web de Aprendizaje, dado que resulta más fácil manejar su contenido en esta etapa, como puede apreciarse en la Figura 4.4.

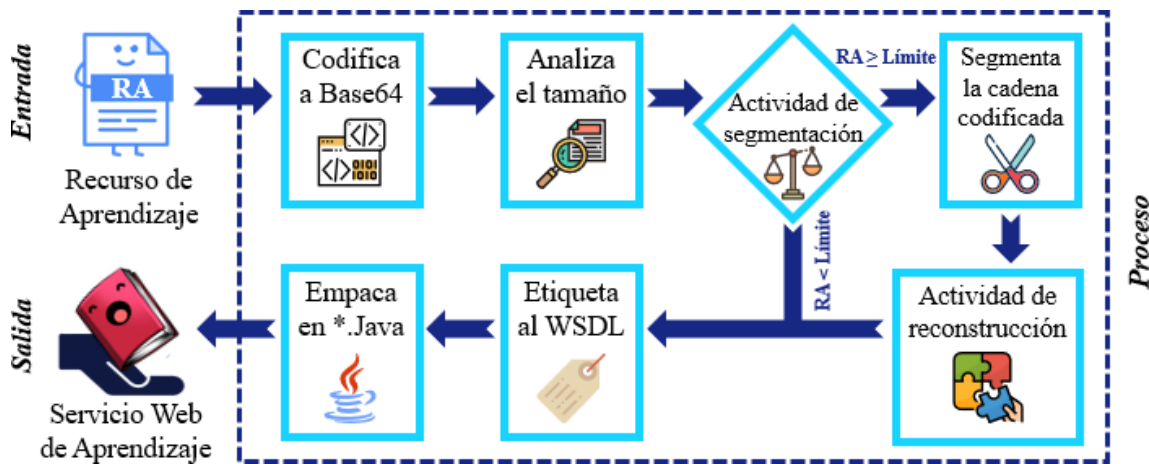


Figura 4.4. Proceso de generación de SWA con la estrategia de gestión

III. Implementación de la Estrategia de Gestión

La modificación por extensión de la herramienta antecedente consistió en aprovechar el patrón de diseño Strategy³ con el cual se desarrolló, realizando los cambios necesarios para que el usuario pueda elegir desde la interfaz de la aplicación, cuál de los dos métodos de generación de Servicios Web de Aprendizaje desea utilizar.

La Figura 4.5 muestra con color aguamarina las clases que fueron modificadas en la herramienta antecedente y de color verde a las que fueron creadas en este trabajo, siendo la clase “UploadBig.java” donde se encuentra implementado el método de generación de SWA mediante la estrategia de gestión definida, y la clase “CompInvocacionRest.Java” para la composición de Servicios Web de Aprendizaje específicamente con servicios de tipo REST.

³ Strategy es un patrón de diseño en el desarrollo de software, clasificado de comportamiento porque determina cómo se debe realizar el intercambio de mensajes entre objetos para resolver una actividad. Permite tener un conjunto de algoritmos donde el cliente elige el que le conviene y puede intercambiar según sus necesidades.

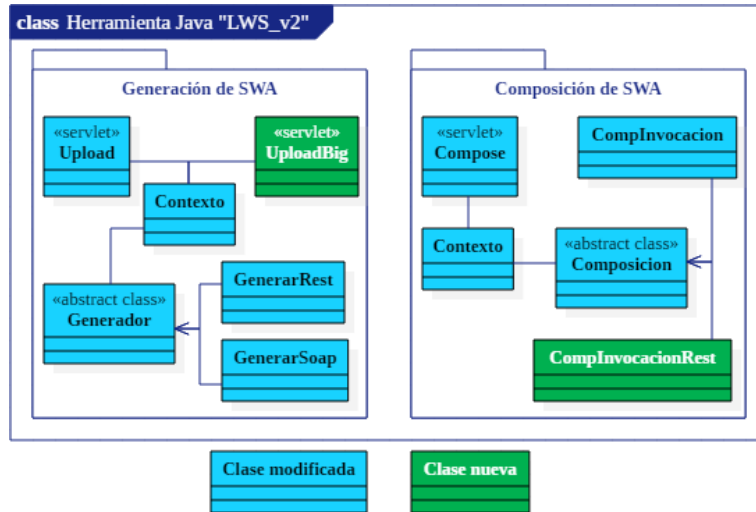


Figura 4.5. Diagrama de clases de la herramienta extendida

Las actividades establecidas en la estrategia de gestión para el proceso de generación de un SWA con contenido, consisten en la segmentación del recurso de aprendizaje en caso de ser necesario y su reconstrucción para poder ser presentado correctamente.

Actividad de Segmentación

La primera actividad consiste en segmentar un Recurso de Aprendizaje (RA) a partir del análisis del tamaño del recurso. Si este es menor a un cierto límite de tamaño, se codifica en Base64 y se empaqueta como servicio web. Pero si el recurso es igual o mayor a este límite de tamaño, se realiza la segmentación en su cadena Base64 resultando tantos segmentos como sean necesarios con un tamaño máximo cada uno, con excepción del último segmento que puede ser de menor tamaño.⁴ La Figura 4.6 presenta el comportamiento de esta actividad.

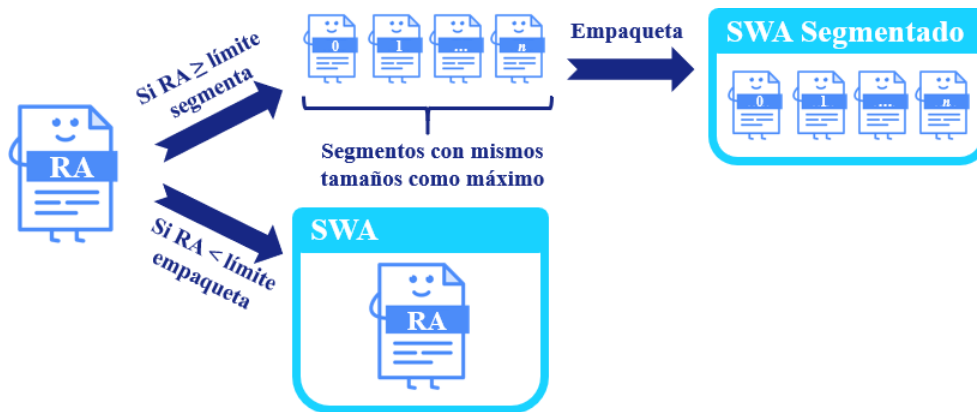


Figura 4.6. Definición del límite de tamaño para realizar la segmentación

⁴ Las medidas asignadas en este trabajo son 8.58 MB en el límite para realizar la segmentación y de 2.86 MB para el contenido de cada segmento como máximo, los cuales fueron definidos con base a los resultados obtenidos en la Prueba Preliminar 1 descrita en el Capítulo 5.

El código que implementa la actividad de segmentación con base en las medidas establecidas se presenta en la Figura 4.7.

```

uploadBig.java x
Source History
103 int len;
104 int fileSize = (int) filePart.getSize();
105 byte[] temp;
106 if (fileSize > (9000000)) { // Regla de segmentación a partir de los 8.58 MB
107     temp = new byte[3000000]; // Regla de asignación de 2.86 MB de tamaño para cada segmento
108 } else {
109     temp = new byte[fileSize];
110 }
111
112 while ((len = in.read(temp)) > 0) {
113     byte[] data;
114     if (len == temp.length) {
115         data = temp;
116     } else {
117         data = new byte[len];
118         System.arraycopy(temp, 0, data, 0, len);
119     }
120     String aux = new String(Base64.getEncoder().withoutPadding().encode(data));
121     encodeFileStr.add(new StringBuilder(aux));
122 }
    
```

Figura 4.7. Código implementando la condición de segmentación

Actividad de Reconstrucción

La segunda actividad consiste en el reordenamiento de los segmentos obtenidos en la actividad de segmentación, con la finalidad de poder reconstruir al recurso de aprendizaje a su estructura original. Para ello, fue necesario modificar las etiquetas del archivo WSDL que se genera en cada SWA en el caso de SOAP, e implementar un arreglo de invocaciones para obtener la cadena segmentada en el orden correcto para su reconstrucción.

La Figura 4.8 se divide en dos partes, en el lado izquierdo (Figura 4.8.a) se muestra la estructura original del archivo WSDL de un Servicio Web de Aprendizaje con contenido. En el lado derecho (Figura 4.8.b) se presenta de manera resumida la modificación del WSDL cuando se realiza la actividad de segmentación, donde se hace uso de etiquetas que redireccionan la “Parte” que hace referencia al número del segmento que va desde 0 hasta *n*.

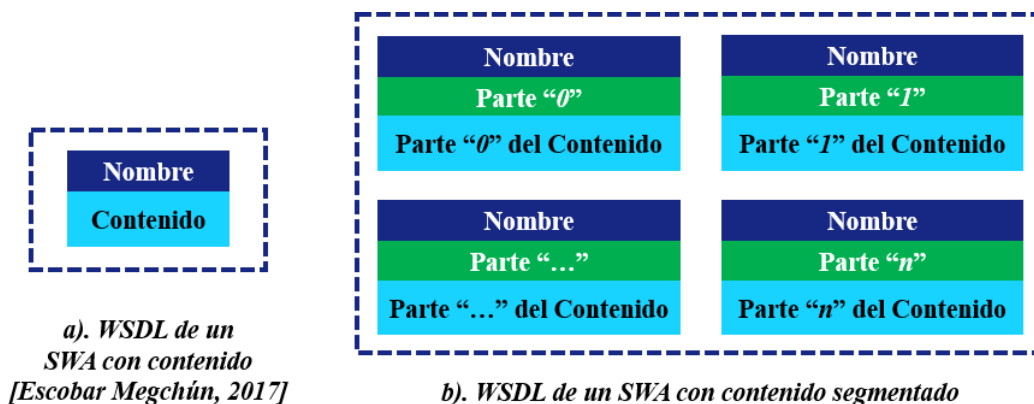


Figura 4.8. Modificación al WSDL de un SWA como actividad de reconstrucción

El archivo WSDL de la Figura 4.8.a cuenta con dos etiquetas, la primera llamada “Nombre” contiene al nombre del recurso de aprendizaje y su formato, la segunda etiqueta “Contenido” que contiene a la cadena Base64 del recurso codificado. De este modo un Servicio Web de Aprendizaje puede entregar un recurso de aprendizaje al usuario. Sin embargo, cuando el tamaño de su recurso es mayor a los 50 MB, este no puede ser presentado debido a que tiene problemas en el almacenamiento dinámico del servidor. Es por ello que debe realizarse una actividad de segmentación en su contenido y una actividad de reconstrucción que permita el correcto ordenamiento en los segmentos resultantes.

La Figura 4.8.b presenta un objeto que resume a la actividad de reconstrucción que consiste en modificar el WSDL agregando un elemento llamado “Parte *n*”, que permite referenciar a los segmentos obtenidos en la actividad de segmentación y modifica al elemento “Contenido” por “Parte *n* del Contenido”, donde el valor de *n* comienza desde 0 siendo este el segmento inicial y termina en *n* correspondiendo al último segmento de un recurso, de tal manera que se tenga el orden correcto en la secuencia para poder reconstruir al recurso contenido en el Servicio Web de Aprendizaje. La forma en que se presenta esta modificación al WSDL se indica en la Figura 4.9, donde aparecen las etiquetas correspondientes a los métodos implementados para el envío de información. La etiqueta “send_archivo” corresponde al nombre del recurso, la etiqueta “send_contenido” corresponde al contenido de un segmento del recurso, y la etiqueta “send_partesrestantes” corresponde al método para analizar si existen más segmentos a unir al momento de reconstruir el recurso.

```

+<!-->
+<!-->
-<definitions targetNamespace="http://Services/" name="TestSoap18">
  <types/>
  +<message name="send_partesrestantes"></message>
  +<message name="send_partesrestantesResponse"></message>
  <message name="send_archivo">
  +<message name="send_archivoResponse"></message>
  +<message name="send_contenido"></message>
  +<message name="send_contenidoResponse"></message>
  <portType name="TestSoap18">
    <operation name="send_partesrestantes">
      <input wsam:Action="http://Services/TestSoap18/send_partesrestantesRequest" message="tns:send_partesrestantes"/>
      <output wsam:Action="http://Services/TestSoap18/send_partesrestantesResponse" message="tns:send_partesrestantesResponse"/>
    </operation>
    <operation name="send_archivo">
      <input wsam:Action="http://Services/TestSoap18/send_archivoRequest" message="tns:send_archivo"/>
      <output wsam:Action="http://Services/TestSoap18/send_archivoResponse" message="tns:send_archivoResponse"/>
    </operation>
    <operation name="send_contenido">
      <input wsam:Action="http://Services/TestSoap18/send_contenidoRequest" message="tns:send_contenido"/>
      <output wsam:Action="http://Services/TestSoap18/send_contenidoResponse" message="tns:send_contenidoResponse"/>
    </operation>
  </portType>
  <binding name="TestSoap18ImplPortBinding" type="tns:TestSoap18">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
    +<operation name="send_partesrestantes"></operation>
    +<operation name="send_archivo"></operation>
    +<operation name="send_contenido"></operation>
  </binding>
  <service name="TestSoap18">
    <port name="TestSoap18ImplPort" binding="tns:TestSoap18ImplPortBinding">
      <soap:address location="http://localhost:8080/TestLWS_SOAP/TestSoap18Impl"/>
    </port>
  </service>
</definitions>

```

Figura 4.9. WSDL de un SWA con contenido segmentado

Para el caso de servicios de tipo SOAP, ocurre la modificación del WSDL donde se generan 3 clases, pudiendo ser más clases como segmentos se hayan generado para el recurso. La clase de la interfaz del SWA, la clase con sus métodos de implementación y al menos una clase con el recurso codificado en Base64. La primera clase es la que lleva el nombre del SWA asignado por el usuario, que corresponde a la interfaz del Servicio Web de Aprendizaje como se muestra en la parte superior de la Figura 4.10, con el nombre “TestSoap18.java” que corresponde al WSDL del servicio indicado en la Figura 4.9. La segunda clase contiene a los métodos de implementación del Servicio Web de Aprendizaje, con los nombres “send_archivo”, “send_partesrestantes”, y “send_contenido” como se muestra en la Figura 4.11.

```

1 package Services;
2
3 import javax.jws.WebService;
4 import javax.jws.WebMethod;
5 import javax.jws.WebParam;
6 import javax.xml.soap.SOAPBinding;
7 import javax.xml.soap.SOAPBinding.Style;
8
9 @WebService
10 @SOAPBinding(style = Style.RPC)
11 public interface TestSoap18 {
12
13     @WebMethod(operationName = "send_archivo")
14     public String send_archivo();
15
16     @WebMethod(operationName = "send_partesrestantes")
17     public int send_partesrestantes(@WebParam(name = "parte") int parte);
18
19     @WebMethod(operationName = "send_contenido")
20     public String send_contenido(@WebParam(name = "parte") int parte);
21
22 }
    
```

Figura 4.10. Interfaz de un Servicio Web de Aprendizaje en SOAP

```

5 @WebService(serviceName = "TestSoap18", endpointInterface = "Services.TestSoap18")
6 public class TestSoap18Impl implements TestSoap18 {
7
8     private String name = "BeginningHTML-XHTML-CSS-JavaScript_Ducket.docx";
9     private final int partes = 5;
10
11     @Override
12     public String send_archivo() {
13         return name;
14     }
15
16     @Override
17     public int send_partesrestantes(int parte) {
18         return ((partes-1) - parte);
19     }
20
21     @Override
22     public String send_contenido(int parte) {
23         if(parte == 0) {return new TestSoap18_parte0().contenido();}
24         if(parte == 1) {return new TestSoap18_parte1().contenido();}
25         if(parte == 2) {return new TestSoap18_parte2().contenido();}
26
27     }
28 }
    
```

Figura 4.11. Métodos de implementación para un SWA en SOAP

Y la tercera clase “TestSoap18_parte0.java”, y en su caso el resto de las clases generadas desde “TestSoap18_parte0” hasta “TestSoap18_parten”, corresponde al contenido de cada segmento de la cadena Base64 del recurso de aprendizaje como se aprecia en la Figura 4.12.

Siendo una clase para cada segmento. En el caso de SOAP, la reconstrucción ocurre mediante la modificación del archivo WSDL del SWA.

```

1 package Services;
2 public class TestSoap18_parte0 {
3
4     public TestSoap18_parte0() {
5     }
6     public String contenido() {
7         StringBuilder sb = new StringBuilder();
8         sb.append("UEsDBBQAAGIAEzqg05nQWRAENUQAORF8QARABEAD29yZC9kb2N1bWVudC54bWxVVA0ABw0zpVvNM6VcDT01X0w923bjuJHv+xWM5zG6");
9         sb.append("qX7MZLGIj5s1lWohnhPCgQABoT5/kPB3iW/oEAMONWjTZUe8wDHzGNLQ8QI1JAl1LM0bifgKc4F2WeDgkjrLBhtM1PaamBRxqRvLVjCj");
10        sb.append("Ine2leYt25xoU2ZT9CKUBjYjwG1rKxAIzrDGD+tdYnyQ104UXvM2u1yKxRddR2Go0htVF37fF27uirklcr0QvPMvxoYwThWvsHr4ned");
11        sb.append("ARFyjQORZErGp5UjwQU8S8rW1CxMhFRJQuU69rMf7t7vPTbn7RMCw7bSbMrDPV84f21e0ZoGWxkd27AaojnFid14+c29wMIQzApYJF");
12        sb.append("KEoi23a4uMybf04p5dw96WfepHJuZyffWR5FLM9KXKd3MdoPn359zuBmYDiRd1DlMh3ecgdQCGO4kCfZIOaf1BqE6mAKRzM4GkOPPJA");
13        sb.append("TvmIDAFnOH+QN9s/Pf0AnXFtLkVtCkJB7h9Sn3xGXDs/eAe2NX3DQt6qBChj9/2p19+TmBiziAvrJI7D+7s7Zk/ollAmgDg8Tz3EwIn");
14        sb.append("EjCumcB9b+7SiR1Hz6s5tLLzype7Hb/aLeOgdHd1uKHbL2DKEM5kPPY9MQns07fq0ghRD0V1SmEqOeiSoU9QwXUqBCinozKFELUUI");
15        sb.append("+72PdGB8JX7wAkTvKIXtArIQ5K5b2BJC+8G+68sUefXI99j19DcpmoMaUqgmBfzvNS3grb4oRj+3jNiseB3bsk2B85YzBUG7+1zLA0");
16        sb.append("sBu0RrRbK2gVL6T1/xAkV0ONuAxfJUA5b0VKVvO4ZtGLPKBRxSoFMMZ/1rW8hGS+TAM7SjwthodTyO4RB+C+2g/97gb3Df5Bgbjw");
17        sb.append("06DjnmOqsSQHwNLFbUFcFSOHEJ9kZJ50KeFfAzR3NIXH8hMVZ2pgrDa9fkyQKI/Mm/J8Mb4eoY8E80IX8TNxwx4YkMQEz+K8AKIo");
18        sb.append("5G74WjEdvrJdy3bdntd7875h6+GfN8koUeFM2Bu+EBELPh0bV1VrL+ZJi3LvpHbVr/dnfJdLYn+9JVjdVbF2kckafBP32VxDOq/hS");
19        sb.append("F610cFd5GejbyD+c7BvM+1DJITZaUWcSU7Vc261+snxjD+vo8+QbJ6uq1G1a58s2vkdhwK0c1IzYc1105FmIzi19Ym6StHRefo4m9");
20        sb.append("bxSZ+/C7j8ztZdp5Oq+eIhez8Pz2abtbnm6yup5rN0xaquEUYRm8WmxJslKR31of/4H3dwPgezqRFhJ0EgUSLmDaj7gTeL5euJCE");
21        sb.append("vbE6/1lQR09GP6twaE9BP6ubq7Wrkat1AFwL2gdX7x+Xj4QRCD0XRLE3Y2tBSR47xboDHHYUfJ44LctCkVbTvc2xi0N0pDCwEE1cpA");
22        sb.append("+ekrIG7WZ/BNic5DuB/eGStufRs2Opet6gFHEP5U4D1gb3rdF0ZV12hiqKvR9Yv3U4LpNP09bQaIZAWfePyfE2TLT9YXcGhH5q1Mh");
23        sb.append("4328A874444T0Z25DN2rBY7G4G3b5h1s0dVbksR7L4YvYvW784+M4E8+2zRmM2/z0MxYd4V071f680136mKZ6zY8NYE7u");

```

Figura 4.12. Segmento del recurso de aprendizaje contenido en un SWA en SOAP

Para el caso de servicios tipo REST, al igual que cuando se genera un Servicio Web de Aprendizaje de tipo SOAP se obtiene una cantidad de clases como segmentos se requieran, se generan 3 clases. Una clase que implementa al SWA y se muestra en la Figura 4.13. Una segunda clase con la configuración para su despliegue como se aprecia en la Figura 4.14, y al menos una clase con el recurso codificado en Base64 como se muestra en la Figura 4.15.

```

16 @Path("TestRest30")
17 public class TestRest30Rest {
18
19     @Context
20     private UriInfo context;
21     private final int partes = 181;
22
23     public TestRest30Rest() {
24     }
25
26     @GET
27     @Produces(MediaType.TEXT_XML)
28     public List<JAXBElement<String>> send_datos(@QueryParam("parte")int parte) {
29         return Arrays.asList(new JAXBElement[]{
30             new JAXBElement(QName.valueOf("archivo"), String.class, name),
31             new JAXBElement(QName.valueOf("contenido"), String.class, contenido(parte)),
32             new JAXBElement(QName.valueOf("partes_restantes"), Integer.class, (partes-1) - parte)
33         });
34     }
35     private String name = "MatematicasAplicadasParaLaVidaReal_Paenza.mp4";
36     public String contenido(int parte) {
37         if(parte == 0) {return new TestRest30_parte0().contenido();}
38
39         if(parte == 1) {return new TestRest30_partel().contenido();}
40
41         if(parte == 2) {return new TestRest30_parte2().contenido();}

```

Figura 4.13. Implementación de un Servicio Web de Aprendizaje en REST

```

1 package Rest;
2
3 import java.util.Set;
4 import javax.ws.rs.core.Application;
5
6 @javax.ws.rs.ApplicationPath("webresources")
7 public class ApplicationConfig extends Application {
8
9     @Override
10    public Set<Class<?>> getClasses() {
11        Set<Class<?>> resources = new java.util.HashSet<>();
12        addRestResourceClasses(resources);
13        return resources;
14    }
15    private void addRestResourceClasses(Set<Class<?>> resources) {
16        resources.add(Rest.TestRest30Rest.class);
17    }
18
19 }
20

```

Figura 4.14. Archivo de configuración para el despliegue de un SWA en REST

Y la tercera clase “TestRest18_parte0.java”, y en su caso el resto de las clases generadas desde “TestRest18_parte0” hasta “TestRest18_parten”, corresponde al contenido de cada segmento de la cadena Base64 del recurso de aprendizaje como se aprecia en la Figura 4.15. Siendo una clase para cada segmento. En el caso de REST, la reconstrucción ocurre mediante los métodos de implementación que conforman al SWA.

```

1 package Rest;
2 public class TestRest18_parte0 {
3
4     public TestRest18_parte0() {
5     }
6     public String contenido(){
7         StringBuilder sb = new StringBuilder();
8         sb.append("UEsDBBQARgAIAEzqzgo5nQWRrAENUQAORF8QARABEAR29yZC9kb2N1bWVudC54bWxVVA0ABw0zpVwNM6VcDT01X0w923bjuJHv+xWMSzG0jQvBibLdI");
9         sb.append("qX7MZLGIWjIsI1WohnhPCgQAB0T5/kPB3iW/ oEAMONWjTzUe8wDHRGNLQ8Q11JA11M0bi fgLKc4F2WeDgkjrLBHtM1PaamBRqgAvLVjCkKeDPALTC");
10        sb.append("In2leYT25xoU2ZT9CkUBjYwG1rKxAlrzDGD+tcDInyQ104UXvM2ulykxRddr2GcohtVF37f27uirklcr0QvPMvxoYwThWvSr4nedZmh/AArVN");
11        sb.append("ARFyjQORZErGP5UjwQ8S8rW1CxMhFRJQuU69rMf7t7vPTbn7RMCw7bSbMrdfP784f21e0ZoGwXkd27AaojnFldl4+c29wMIQzApYsJFW3vEh1U0w/");
12        sb.append("KEoI23a4uMybf04p5dw96WFepHJuZyFwR5F1M9KOKd3MdoPn359zuBmYDlRqLD1Mh3scgdCQGO4kCfZIOaf1BqE6mAKR2M4GKOPPPJAGvSKJLEkM");
13        sb.append("TvmIDANOH+QNSs/Pf0AanXFtLKVTckJB7h9Sn3xGXDs/ eAe2NX3Qct6qBChj9/2p19+TmBz1AvrJI7D+7s7Zk/ oLLAmgDg8Tz3EwIn52+jhTkrh");
14        sb.append("EjCumc8Bb+7SiR1Hz6eStLLzype7Hb/ sLeOGdFHD1uKHbL2DKEM5kPFY9MQns07fq0ghRD0V1SmEqQe1s0UQ9WxUqBcInozKFLUUI1GhQoh6M4pTY");
15        sb.append("+72PdG8BJ7wRkTvkIXITArIQ5K8zBJJC+8G+68stxEXI99j19DcpmoMaUqgmBfzvNS9grb4cRj+3jNiSeB3bsk2B85Y5zBUG7+1z1A03RC56nR3g");
16        sb.append("sBu0RrrBkN2qVL6T1/xAkV00NuAxmfJUa5b0VKVv04ZtGLPKRxs0FMMZ/1rW9ShGS+TAm7SjwthodTy04RB+C+2g/97gb3Df5B8BjwrfQd43GI");
17        sb.append("O6DjnMoQsSQHwNLFbUFCFS0HEJ9kZJ50KeFFAzR3NIXH8hMVZ2pgrDa9fkyQKII/Mm/J8Mb4eoY8E80IX8TNxwxw4YkMQE2+KH8AkIoTOLXC3QW6");
18        sb.append("5G74WjEdvrJdy3bdntnD787Sh6+Gfn8koUeFM2Bu+EBELPh0Bv1rL+2Ji3LvpHbVr/ dnfJdLYn+9JVj dVbfZkckfabDP32VxDOq/hSZL36bG3d5");
19        sb.append("F610cFd5GejbyD+c7BvM+LDJITZaUwCSU7VcZ61+snxjD+vo8+Qbj6uq1G1a58s2vkdhwKoc1IzYoCi105Fm8Izi19YM6StHREfo4m9as444Evl1V");
20        sb.append("bXSZ-/C7j8ztZbp5Oq+eihez8Pz2abtbmnw6yup5rNOxaqwuEUYRm8WmxJslKR31of/4H3dwPgezqRfHJ0EgUSLMDaj7gTeL5euJCEuf4WreW02");
21        sb.append("vbe6/1QR09GP6twaE9BP6ubq7Wkat1Afw1L2gdX7x+Xj4QRCD0XRLE3YZtBSR47XboDHYuFU44LCTcKvBtvc2xi10NpDCWEE1cpACWDV8f299");
22        sb.append("+ekrIG7WZ/BNIC5DuB/ eG8tuf8s20pet6gFHEP5U4D1gb3rdFoZV12hiqKVz9rYv3U4LpNPo9bQaIZAWfePyfE2TLT9yXCgHnh5q1Mhr8uXz37/2");
23        sb.append("42RAOfZjn6jI0A7CSDD2CBxX2GgVgPFlaRoVVkaBU2WoXpUDWZj+zmj8Ez2zHmpN+Z/nAmoxUdVq71fa801RequK0g2rGNSXfYlmQ8z2EM8uB");

```

Figura 4.15. Segmento del recurso de aprendizaje contenido en un SWA en REST

Cabe señalar, que las clases de los segmentos indicadas en la parte superior de la Figura 4.12 para SOAP y en la parte superior de la Figura 4.15 para REST, contienen el conjunto de líneas de la cadena Base64 del recurso de aprendizaje que se corta al llegar a los 2.86 MB y continúan en las siguientes clases correspondientes a los demás segmentos, hasta obtener toda la cadena del recurso de aprendizaje codificado en Base64.

Estas actividades de segmentación y reconstrucción pueden apreciarse de manera más clara en un diagrama BPMN para la generación de un Servicio Web de Aprendizaje con contenido, como se indica en la Figura 4.16, generado con Enterprise Architecture.

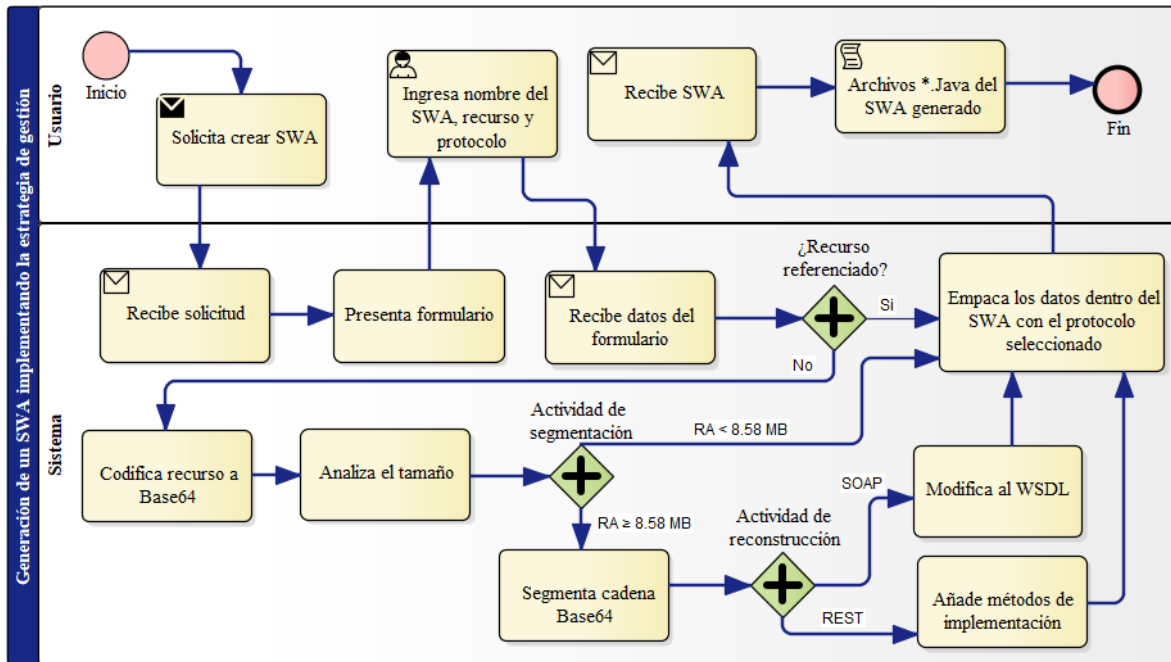


Figura 4.16. Diagrama BPMN para la generación de un SWA implementando la estrategia de gestión

El usuario solicita al sistema la creación de un SWA donde le asigna el nombre, el tipo del recurso de aprendizaje y su protocolo de transferencia con un formulario. El recurso puede ser referenciado o no referenciado, el primero es cuando se encuentra en Internet y no se tiene posesión por lo cual no puede realizarse la conversión a Base64, y el segundo es cuando el usuario ingresa al recurso de aprendizaje desde su dispositivo. Los protocolos de transferencia pueden ser SOAP o REST. Si el recurso es no referenciado se codifica a Base64 y se analiza su tamaño para realizar la actividad de segmentación. Cuando el recurso es igual o mayor a los 8.58 MB segmenta la cadena Base64 en tantos segmentos como sean necesarios y aplica la actividad de reconstrucción, modificando al WSDL para ordenar los segmentos. Posteriormente los datos son empaquetados dentro de un servicio web con el protocolo seleccionado y el sistema le entrega al usuario los archivos Java generados del Servicio Web de Aprendizaje con contenido.

La manera en que se genera y se presenta un Servicio Web de Aprendizaje con contenido se detalla brevemente en la Figura 4.17. Donde un usuario interactúa en la interfaz de la herramienta Java, cargando un Recurso de Aprendizaje y eligiendo el tipo del servicio a generar. La herramienta codifica al recurso a Base64 y analiza su tamaño, si este es igual o mayor a los 8.58 MB lo segmenta en tantas partes como sea necesario, finalmente empaqueta los datos dentro de un servicio web que es entregado al usuario. Posteriormente, el usuario carga el SWA generado a un servidor de aplicaciones para ser desplegado y poder consumirse por el cliente PHP. En caso de que el recurso contenido en el servicio web se encuentre segmentado, el cliente se encargará de reconstruirlo de acuerdo a la secuencia de sus

segmentos para poder visualizarlo de manera correcta en la vista del navegador web al usuario.

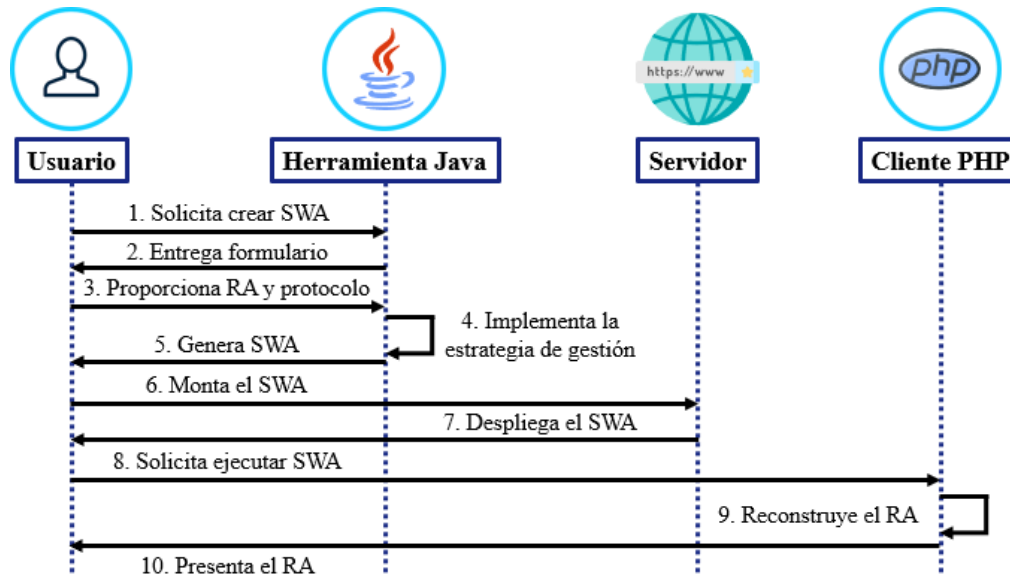


Figura 4.17. Diagrama de secuencia para generar y presentar un SWA

Características para su segmentación y reconstrucción automática

En este trabajo se identificaron las características⁵ necesarias para realizar la segmentación y la reconstrucción automática de un Recurso de Aprendizaje que se encuentra empaquetado como un Servicio Web de Aprendizaje, como son:

1. Límite para realizar la segmentación:

Es un tamaño estándar que limita a partir del cual se realizará la segmentación del Recurso de Aprendizaje.

2. Tamaño de los segmentos del recurso:

Es una medida definida para el tamaño máximo que puede contener cada segmento.

3. Método de reconstrucción:

Es la forma en que se ordenarán los segmentos resultantes para reconstruir al recurso a su forma original.

4. Validar el contenido de los segmentos:

Deben validarse los contenidos en cada uno de los segmentos del recurso, con la finalidad de verificar que no existan daños en su infraestructura.

⁵ En este trabajo el límite asignado es 9,000,000 B (8.58 MB), el tamaño para cada segmento es 3,000,000 B (2.86 MB), el método de reconstrucción es mediante la modificación del WSDL en el caso de SOAP y añadiendo elementos a los métodos de implementación que conforman al SWA en el caso de REST, y la validación en el contenido se dio al solventar la aparición del código blanco en los segmentos de los recursos. Dichas características fueron identificadas en la realización de la Prueba Preliminar 1 descrita en el Capítulo 5.

Guía de Composición

Para la generación de un Servicio Web de Aprendizaje Compuesto es necesario conocer una serie de reglas que permitan su correcta ejecución y presentación, por lo que se debe contar con una guía de composición que facilite su realización.

El proceso de composición ocurre de una manera semi automática ya que requiere la intervención del usuario para la selección de los Servicios Web de Aprendizaje y su orden de presentación, el sistema generará un arreglo de invocaciones con estos y los empaquetará como un Servicio Web de Aprendizaje Compuesto (SWA-C). Dicho proceso puede observarse de manera resumida en la Figura 4.18.

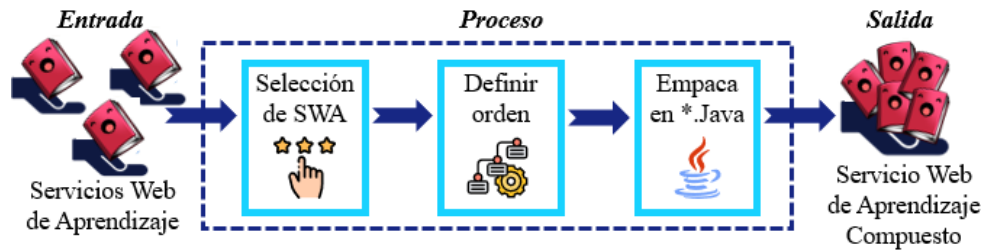


Figura 4.18. Proceso de composición de SWA en la herramienta antecedente [Escobar Megchún, 2017]

Para una buena práctica en la generación de Servicios Web de Aprendizaje Compuestos se define una serie de reglas a considerar, las cuales se describen en la Figura 4.19.

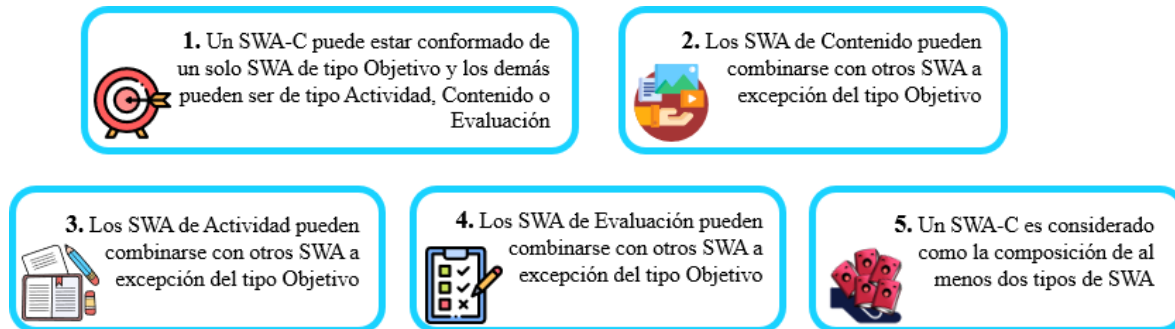


Figura 4.19. Reglas de composición [Escobar Megchún, 2017]

Para propósitos de rendimiento que mejoren los tiempos de respuesta al ejecutar un SWA-C, en este trabajo de investigación se define una recomendación: **“Un SWA-C debe componerse de SWA del mismo tipo de servicio”**.⁶ Debido a resultados obtenidos en pruebas de composición, donde al presentar Servicios Web de Aprendizaje Compuestos y alguno de sus recursos contenidos fuera de 90 MB o superior y de diferente tipo, es decir SOAP y REST, estos presentaron mayores tiempos de respuesta y no llegaron a visualizarse correctamente. Por lo que durante la generación del Servicio Web de Aprendizaje compuesto deben seleccionarse servicios del mismo tipo. Esto se indica en la Figura 4.20.

⁶ El tipo de servicio se refiere al protocolo o la arquitectura de transferencia de datos, pudiendo ser entre SOAP o REST. El tipo de SWA se refiere al tipo de Objeto de Aprendizaje al cual pertenece el Recurso de Aprendizaje que contiene, pudiendo ser entre: Objetivo, Contenido, Actividad o Evaluación.

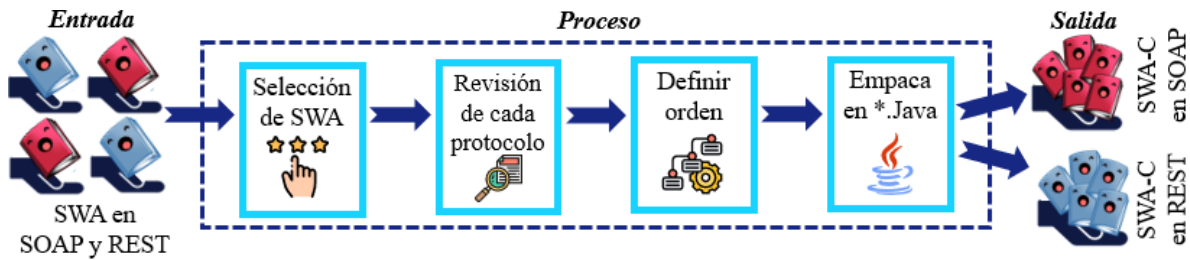


Figura 4.20. Proceso de composición de SWA mediante separación de protocolos

El Servicio Web de Aprendizaje Compuesto generado por la herramienta contendrá el arreglo de invocaciones de los SWA en el orden establecido por el usuario, los cuales serán desplegados mediante proxis dinámicos en caso de pertenecer a SOAP o a través de conexiones HTTP en caso de ser en REST.

El proceso de composición de SWA mediante la separación de protocolos, puede apreciarse de manera más clara en un diagrama BPMN como se indica en la Figura 4.21, generado con Enterprise Architecture 12.1.1227 (Build 1227).

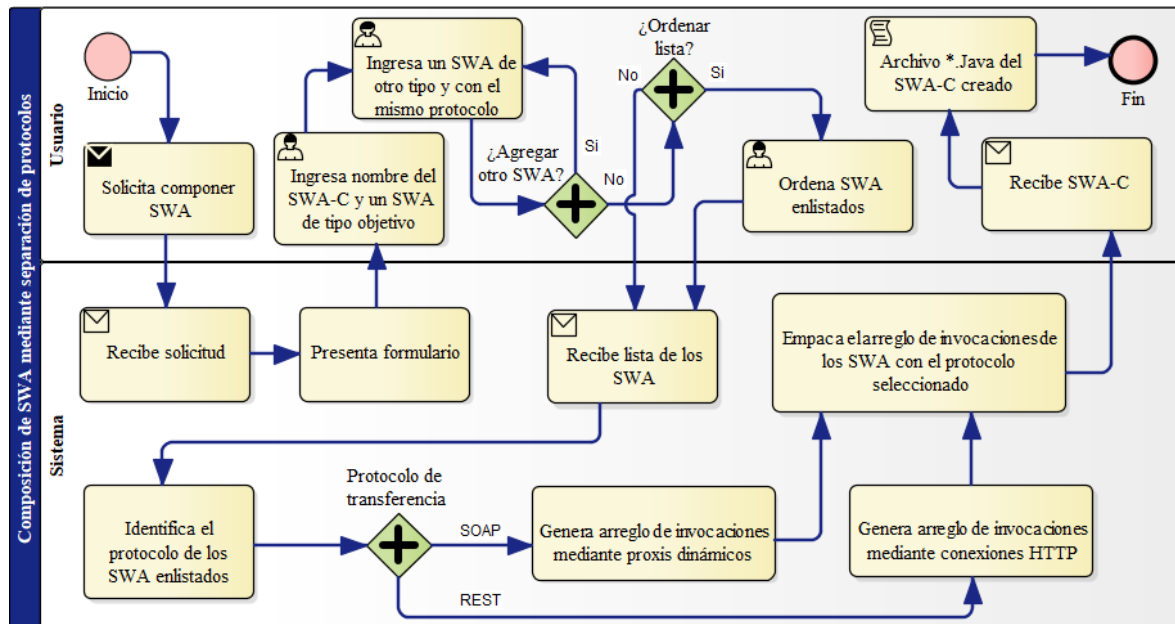


Figura 4.21. Diagrama BPMN para la composición de SWA separándolos por sus protocolos

Al proceso de composición descrito en la Figura 4.21, se incorpora la revisión del protocolo para que todos los Servicios Web de Aprendizaje a componer pertenezcan al mismo tipo de servicio. El usuario ingresa un SWA de tipo Objetivo con un tipo de servicio, sea SOAP o REST, define el nombre del SWA-C a generar e ingresa el resto de SWA con los mismos tipos de servicio del primero entre los cuales pueden ser de los tipos Contenido, Actividad o Evaluación. Una vez teniendo la lista de los SWA en el orden deseado se enviará la petición para realizar su composición, el sistema identificará el tipo de los servicios enlistados y generará un arreglo de invocaciones empaquetado en un archivo Java como un SWA-C.

Características para su Composición Automática

Este trabajo de investigación tiene como objetivo identificar las características necesarias para una guía de composición automática, para lograrlo, el sistema debe reconocer la información en cada recurso de aprendizaje y enviarlo a un proceso de composición de secuencias lógicas que atiendan a objetivos de aprendizaje, lo cual no se encuentra dentro del alcance ni la problemática a tratar.

Las características identificadas para que el sistema realice la composición de manera automática requieren de la intervención del factor humano, es decir, son aquellas que realiza el usuario desde la interfaz de la aplicación, como son:

1. Selección de los SWA a componer:

El sistema debe buscar e identificar a los Servicios Web de Aprendizaje candidatos a ser compuestos.

2. Ordenar a los SWA:

El sistema debe identificar los contenidos en cada recurso, de tal manera que ordene a los servicios de acuerdo a la secuencia requerida.

3. Nombrar al SWA-C:

El sistema debe generalizar el concepto de los Servicios Web de Aprendizaje a componer para nombrar al Servicio Web de Aprendizaje Compuesto con base al contenido en sus recursos.

Estas características previamente descritas para la composición automática de Servicios Web de Aprendizaje, pueden servir como base para continuar con este trabajo de investigación en un trabajo futuro.

Capítulo

5

Pruebas & Resultados

En este capítulo se describen las pruebas realizadas para verificar la correcta implementación de la metodología de solución donde se realizan los procesos de segmentación, reconstrucción, presentación y composición como parte de la estrategia de gestión definida para los Recursos de Aprendizaje contenidos en Servicios Web de Aprendizaje, para posteriormente analizar los resultados obtenidos.

5.1. Pruebas Preliminares

Se realizaron pruebas preliminares durante el desarrollo de la herramienta, las cuales permitieron identificar aspectos importantes a emplear como el tamaño estándar que deben tener los segmentos de un recurso contenido en un SWA para su correcta reconstrucción. Estas pruebas preliminares se enlistan en la Figura 5.1.

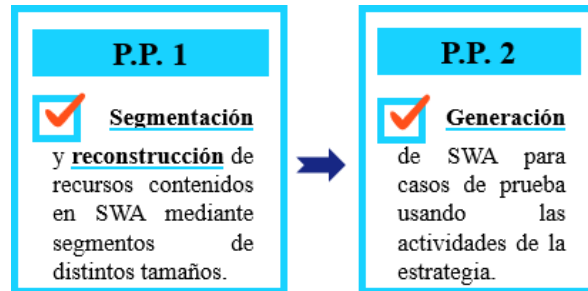


Figura 5.1. Lista de pruebas preliminares realizadas

5.1.1. Prueba Preliminar 1: “Segmentación y reconstrucción de recursos contenidos en Servicios Web de Aprendizaje mediante segmentos de distintos tamaños”

El objetivo en esta prueba preliminar es verificar la correcta reconstrucción del recurso de aprendizaje contenido en un Servicio Web de Aprendizaje. Esto se realizó durante la fase de desarrollo, tanto para SOAP como para REST.

Esta prueba consistió en generar SWA con contenidos segmentados, para definir rangos de tamaños de recursos que van desde los 0.47 MB hasta los 8.58 MB para la cantidad máxima que puede contener cada segmento. Debido que, al reconstruir el recurso con ciertos valores se ocasione la aparición de código blanco. El código blanco es el relleno automático con “0” en una cadena binaria.⁷ Esto ocurre porque se da una mala conversión entre un esquema de codificación a otro. Por ejemplo, un recurso de aprendizaje de manera digital se encuentra en el esquema de codificación ASCII, el cual puede ser un archivo PDF que tenemos en la computadora. Pero cuando queremos que ese recurso sea visualizado en un navegador a través de un servicio web, este debe encontrarse en el esquema Base64.

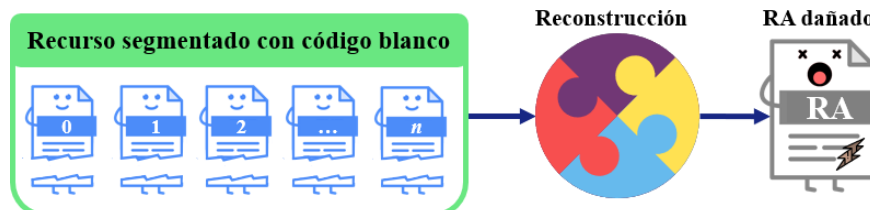


Figura 5.2. Reconstrucción no exitosa de un recurso segmentado con código blanco



La Prueba Preliminar 1 se detalla en la **Página 75**, que pertenece al **Anexo A**.

⁷ La generación de “huecos” o “código blanco” es el relleno automático con ceros (0) ocasionada por un desplazamiento a nivel de operadores de bits. El desplazamiento es ejemplificado en la [Sección 2.15](#).

Es por tal motivo, que se definieron rangos de tamaño para establecer a partir de qué límite se lleva a cabo la segmentación, así como la medida estándar para cada segmento del recurso.

Los resultados en esta prueba preliminar, mostraron que el tamaño en bytes debe ser un número divisible entre 3, para que ocurra de manera correcta la conversión del esquema ASCII a binario, y de binario al esquema Base64.

En el [Anexo A](#) se presentan los 8 rangos definidos para esta prueba, de los cuales solo en 4 de ellos la reconstrucción ocurre de manera exitosa tanto en SOAP como en REST, aunque con diferentes tiempos de reconstrucción, lo cual se muestra en la Figura 5.3.

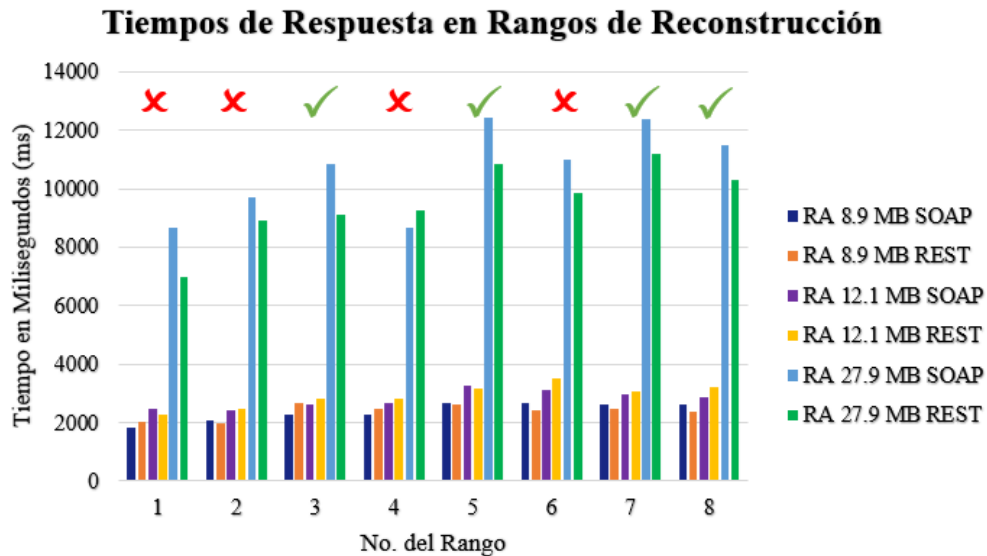


Figura 5.3. Comparación en los tiempos de respuesta en los rangos de reconstrucción

De los rangos presentados en la Figura 5.3, se descartaron los rangos 1, 2, 4 y 6 porque generaban código blanco. Quedando como posibles límites de tamaño a elegir entre los rangos 3, 5, 7 y 8, de los cuales se optó por seleccionar el que presentó menor tiempo de respuesta para reconstruir a los recursos de aprendizaje. Por tal motivo se identificó al rango 3 como el que define el tamaño de 8.58 MB a partir del cual se realice la segmentación de un recurso y de 2.86 MB como el tamaño contenido en cada segmento como máximo.

5.1.2. Prueba Preliminar 2: “Generación de Servicios Web de Aprendizaje para casos de prueba usando las actividades de la estrategia”

El objetivo en esta prueba preliminar es verificar que la estrategia de gestión implementada en la herramienta se efectúe de manera correcta. Esto se realizó durante la fase de desarrollo, tanto para SOAP como REST.

Se consideran como criterios de éxito al realizarse de manera correcta cada una de las actividades de la estrategia de gestión. Donde todos los recursos de aprendizaje ingresados se codifican al esquema Base64 y se empaquetan como un Servicio Web de Aprendizaje. Pero únicamente se segmentan los que son de igual o mayor tamaño a 8.58 MB.



La Prueba Preliminar 2 se detalla en la [Página 78](#), que pertenece al [Anexo A](#).

De todos los recursos de aprendizaje ingresados, todos cumplieron con las condiciones de la actividad de segmentación. Teniendo como resultado que los recursos menores a los 8.58 MB fueran empaquetados como SWA sin ser segmentados. Donde en todos los casos también se cumplió con la actividad de reconstrucción, empaquetando a cada segmento como una clase referenciada al número que le corresponde de acuerdo al orden de la cadena Base64 original del recurso. Y en el caso de los que no fueron segmentados, se creó una única clase que contiene toda la cadena.

La lista de los Servicios Web de Aprendizaje generados aplicando la estrategia de gestión se encuentran enlistados en el [Anexo A](#).

5.2. Ejecución de Pruebas

En esta sección se describen los tipos de pruebas realizados para verificar la correcta implementación de la metodología de solución donde se realizan las actividades de segmentación y reconstrucción de la estrategia de gestión, para la generación y composición de Servicios Web de Aprendizaje con contenido, para posteriormente analizar los resultados obtenidos.

5.2.1. Objetivos de las Pruebas

Los objetivos de las pruebas en este trabajo son:

1. Verificar la generación de Servicios Web de Aprendizaje con contenido mediante las actividades de segmentación y reconstrucción de la estrategia de gestión propuesta en este trabajo.
2. Verificar la ejecución y presentación de los Servicios Web de Aprendizaje que se encuentran segmentados para poder ser visualizados de manera correcta.
3. Verificar la ejecución y presentación para visualizar múltiples recursos de aprendizaje que han sido segmentados y reconstruidos, empaquetados como un Servicio Web de Aprendizaje Compuesto.
4. Comparar los tiempos de respuesta en la entrega del recurso contenido en los Servicios Web de Aprendizaje para el protocolo SOAP y la arquitectura REST.

5.3. Procedimiento de Pruebas

Para llevar a cabo el procedimiento de pruebas, se realizó un plan donde se describen los elementos considerados y la razón por la cual fueron seleccionados, una descripción donde se detallan las vistas de las interfaces usadas en la herramienta y el cliente desarrollados, y un listado de las pruebas realizadas para su debida ejecución, dicho procedimiento se ilustra en la Figura 5.4.

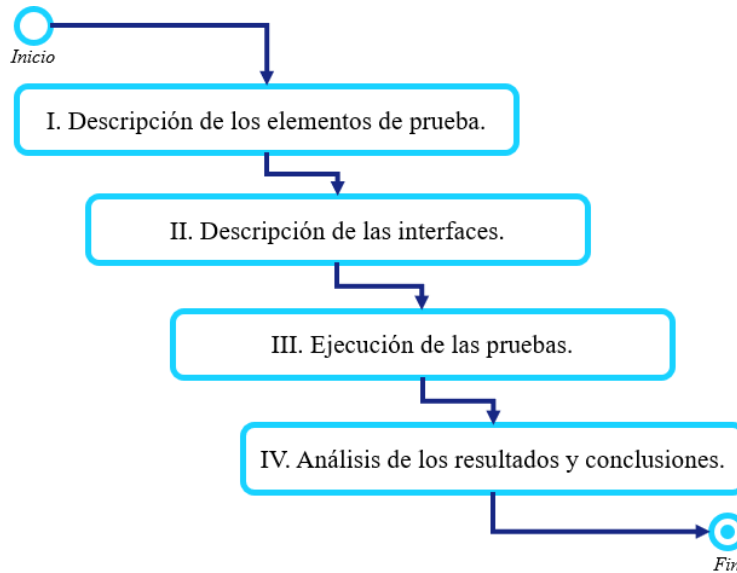


Figura 5.4. Procedimiento de pruebas y resultados

I. Descripción de los Elementos de Prueba

Para la generación de Servicios Web de Aprendizaje se requiere contar con recursos de aprendizaje, los cuales pueden ser de distintos tamaños, formatos como PDF, JPG, DOC, etc., y tipos: Objetivo, Contenido, Actividad y/o Evaluación. Los elementos anteriores se requieren para generar Servicios Web de Aprendizaje, así mismo, como se representa en la Figura 5.5, se requiere contar con repositorios para buscar los recursos de aprendizaje principalmente de gran tamaño, como por ejemplo mayores a los 9 MB, para luego realizar la segmentación y reconstrucción de esos recursos en un entorno de pruebas el cual permita la obtención de los tiempos de respuesta.



Figura 5.5. Elementos de prueba

Repositorios de Recursos

Para la obtención de recursos de aprendizaje, se realizaron búsquedas entre diversos repositorios con contenidos educativos o cursos formados o construidos como objetos de aprendizaje. La lista de los repositorios consultados se presenta en la Tabla 5.1.

Tabla 5.1. Repositorios de recursos

Nombre del Repositorio	Proveedor del Repositorio	Tipos de Recursos	Acceso	Dirección del Repositorio
Biblioteca Digital de la BNP	Biblioteca Nacional del Perú (Lima, Perú)	Repositorio digital de libros	Libre	http://bdigital.bnp.gob.pe
Libros UNAM OpenAccess	Universidad Nacional Autónoma de México (Ciudad de México, México)	Repositorio de libros	Libre	http://www.librosoa.unam.mx
Repositorio Digital	Universidad Central del Ecuador (Quito, Ecuador)	Repositorio de libros	Libre	http://www.dspace.uce.edu.ec
Biblioteca Digital Icesi	Universidad Icesi (Cali, Colombia)	Repositorio de objetos de aprendizaje	Libre	https://repository.icesi.edu.co
University of Nottingham	University of Nottingham (Nottingham, UK)	Repositorio de objetos de aprendizaje	Libre	https://www.nottingham.ac.uk
YouTube	Google (California, USA)	Servicio de alojamiento de vídeos	Libre	https://www.youtube.com

Recursos de Aprendizaje

Posteriormente a la búsqueda en repositorios, se seleccionaron 30 recursos que varían en sus tipos de objeto de aprendizaje, tamaño y formato. Se toma el tamaño como atributo principal para su selección, obteniendo en su mayoría a recursos de tipo Contenido y con formatos PDF y MP4. Para contar con una mayor cantidad recursos con formatos DOC y PPT, se manipularon recursos con la finalidad de duplicar sus contenidos e incrementar en tamaño, obteniendo ahora un total de 39 recursos, como puede apreciarse en la Tabla 5.2.

Tabla 5.2. Lista de recursos de aprendizaje seleccionados

No.	Nombre del Recurso de Aprendizaje	Tamaño	Formato	Tipo
1	Objetivo_ClasesDePiano	0.1 MB	JPG	Objetivo
2	Objetivo_ClasesDePiano	0.2 MB	PNG	Objetivo
3	Objetivo_AlcanosAlquenosAlquinos	0.3 MB	PDF	Objetivo
4	ArduinoManualProgramacion_RuizGutierrez	0.8 MB	PDF	Actividad
5	TeoriaInversion_BuenaventuraGuillermo	1.8 MB	PDF	Evaluación
6	HerramientaGestionEstrategica_RamirezFranco	3.5 MB	PDF	Contenido
7	ComoHacerDiapositivas_Nieto	4.2 MB	PPT	Contenido
8	AssemblyLanguageComputerArchitecture_Detmer	5.1 MB	PDF	Contenido
9	FundamentosBasesDeDatos_Silberschatz	8.9 MB	PDF	Contenido
10	Objetivo-PresentacionCienciasNaturales	9.0 MB	MP4	Objetivo
11	MicrocontroladoresProgramacionBasic_Reyes	9.1 MB	PDF	Contenido
12	BeginningSpringFramework2_Velde_Snyder	9.5 MB	PDF	Contenido
13	Objetivo-PruebaCienciasNaturales	9.9 MB	MP4	Objetivo
14	DataCultureKeynote-ExecTrack_Woodward	10.6 MB	PDF	Contenido
15	CiscoCompleteReference_BrianHill	10.8 MB	PDF	Contenido
16	BeginningHTML-XHTML-CSS-JavaScript_Ducket	11.4 MB	PDF	Contenido
17	CompleteReferenceJava2_HerbertSchildt	12.1 MB	PDF	Contenido
18	ComoHacerDiapositivas_Nieto_(Dup1)	12.6 MB	PPT	Contenido

19	BeginningSpringFramework2_Velde_Snyder	13.8 MB	DOC	Contenido
20	CursoAvanzadoMicrocontroladores_Electronica	14.6 MB	PDF	Contenido
21	ComoHacerDiapositivas_Nieto_(Dup2)	16.8 MB	PPT	Contenido
22	AprendeComoTocarPiano-1raEd	17.1 MB	PDF	Contenido
23	BeginningSpringFramework2_Velde_Snyder_(Dup1)	18.3 MB	DOC	Contenido
24	ComoHacerDiapositivas_Nieto_(Dup3)	25.2 MB	PPT	Contenido
25	BeginningSpringFramework2_Velde_Snyder_(Dup2)	27.4 MB	DOC	Contenido
26	DosCuentos_VilloroJuan	27.9 MB	MP3	Contenido
27	CursoAlgebra_LeyesBasicas	32.5 MB	MP4	Contenido
28	BeginningSpringFramework2_Velde_Snyder_(Dup3)	42.7 MB	DOC	Contenido
29	ComoHacerDiapositivas_Nieto_(Dup4)	50.4 MB	PPT	Contenido
30	BeginningSpringFramework2_Velde_Snyder_(Dup4)	61.1 MB	DOC	Contenido
31	ComoHacerDiapositivas_Nieto_(Dup5)	67.2 MB	PPT	Contenido
32	PianoParaPrincipiantes_TaniaBrou	78.2 MB	MP4	Actividad
33	OrganizacionArquitecturaComputadores_Stallings	95.6 MB	PDF	Contenido
34	MicroprocesadoresIntel_BreyBarry	98.8 MB	PDF	Contenido
35	AlcanosAlquenosAlquinos_QuimicaBasica	133.5 MB	MP4	Contenido
36	IngenieriaDelSoftware_Pressman	293.4 MB	PDF	Contenido
37	AlgebraLineal_Grossman	314.0 MB	PDF	Contenido
38	Documental-FisicaCuantica-ConocimientoOriental	409.9 MB	MP4	Contenido
39	MatematicasAplicadasParaLaVidaReal_Paenza	528.3 MB	MP4	Contenido

Entorno de Pruebas

Se utilizó la herramienta de gestión de pruebas Apache JMeter 5.1.1. De acuerdo a [Apache, 2019] esta herramienta proporciona un entorno de procesos aislados para medir los tiempos de respuesta en las funcionalidades implementadas, y que permite realizar pruebas unitarias de carga y de rendimiento entre distintos tipos de aplicaciones, servidores y protocolos, entre los cuales se encuentran la tecnología de los servicios web y los protocolos SOAP y REST. Se crearon diversas pruebas para cada uno de los procesos; la generación, segmentación, reconstrucción, presentación y composición de SWA tanto atómicos como compuestos⁸. La interfaz de este entorno y las pruebas utilizadas se aprecian en la Figura 5.6.

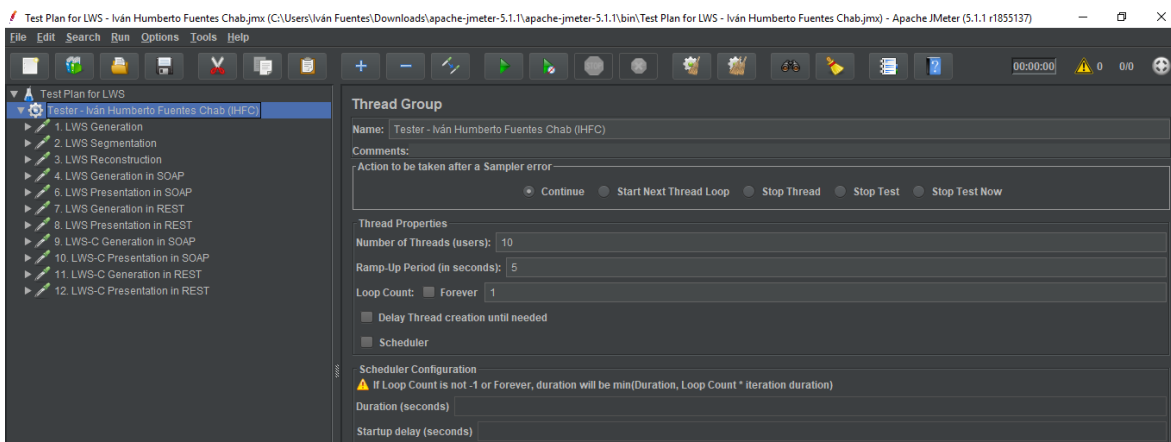


Figura 5.6. Entorno de pruebas Apache JMeter

⁸ Un Servicio Web de Aprendizaje (SWA) es atómico porque solo contiene un recurso de aprendizaje, mientras que un Servicio Web de Aprendizaje Compuesto (SWA-C) contiene al menos dos recursos de aprendizaje.

Este entorno de pruebas permite obtener un promedio estándar de cada funcionalidad en sus tiempos de respuesta, para ello se deben declarar un conjunto de variables como los establecidos en la Figura 5.6, que son:

- **Number of Threads (users):** se declara un número de hilos (usuarios), que será el total de veces que se ejecutará la prueba.
- **Ramp-Up Period (in seconds):** se declara un lapso de tiempo para la ejecución entre cada hilo.
- **Loop Count:** se declara si se desea tener un total de ciclos de repetición, o se marca la casilla “Forever” si se desea que el ciclo de repetición sea infinito hasta su detención por el usuario.
- **Delay Thread creation until needed:** se declara si se desea retrasar la creación de nuevos hilos, hasta que los hilos anteriores hayan finalizado.

En este trabajo, se estableció que cada prueba tenga una ejecución de 10 usuarios en un lapso de 5 segundos, para tener una aproximación más acertada entre los tiempos de respuesta para cada prueba.

II. Descripción de las Interfaces

Para la realización de las pruebas se requirió de dos sistemas independientes, una herramienta Java encargada de la generación mediante segmentación y la composición de Servicios Web de Aprendizaje, y un cliente PHP para la reconstrucción y presentación de los recursos de aprendizaje empaquetados en cada SWA, los cuales se mencionan a continuación y pueden verse simplificados en el mapa de navegación de la Figura 5.7, generado con StarUML 3.1.0.

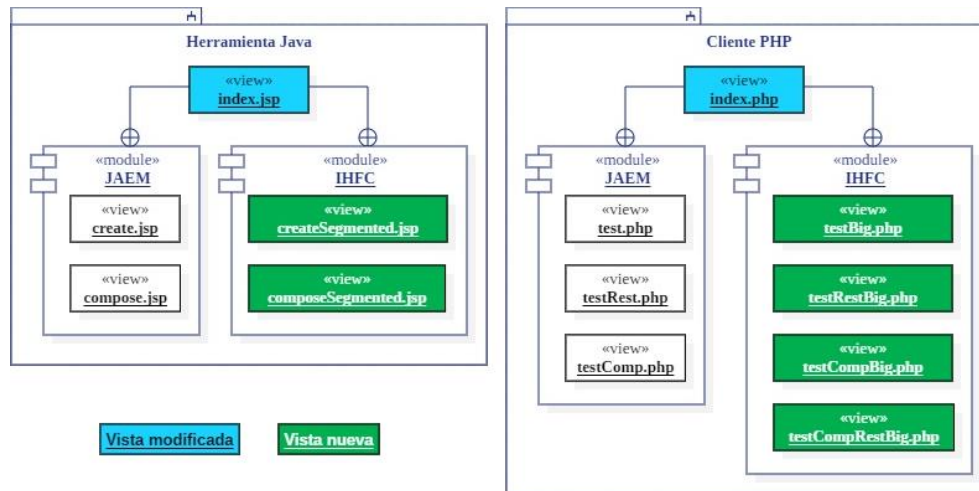


Figura 5.7. Mapas de navegación para las interfaces de la herramienta Java y el cliente PHP

Herramienta Java: “Generación y Composición de Servicios Web de Aprendizaje”

La herramienta Java se encarga de la generación de Servicios Web de Aprendizaje mediante la segmentación de los recursos de aprendizaje empaquetados en estos. De igual manera, la

herramienta permite realizar la composición de múltiples SWA para poder generar a un Servicio Web de Aprendizaje Compuesto, el cual contenga un arreglo de invocaciones de al menos dos SWA cuyos contenidos han sido segmentados. De acuerdo al mapa de navegación, las vistas corresponden a las siguientes funciones:

- **“index.jsp” (Vista):** es la ventana inicial de la herramienta, que presenta el título y los nombres de los autores.
- **“JAEM” (Módulo):** es el módulo que contiene a la herramienta antecedente “LWS” [Escobar Megchún, 2017].
 - **“create.jsp” (Vista):** es la ventana encargada de la generación de un Servicio Web de Aprendizaje a partir de un recurso de aprendizaje.
 - **“compose.jsp” (Vista):** es la ventana encargada de realizar la composición de al menos dos Servicios Web de Aprendizaje.
- **“IHFC” (Módulo):** es el módulo que contiene a la extensión implementada en este trabajo.
 - **“createSegmented.jsp” (Vista):** es la ventana encargada de la generación de Servicios Web de Aprendizaje a partir de un recurso de aprendizaje, donde se aplica la estrategia de gestión para empaquetar recursos codificados en servicios web y segmentarlos en caso de ser necesario.
 - **“composeSegmented.jsp” (Vista):** es la ventana encargada de realizar la composición de al menos dos Servicios Web de Aprendizaje cuyos recursos empaquetados han sido segmentados.




Las interfaces desarrolladas en la Herramienta Java se detallan en la **Página 81**, correspondiente al **Anexo B**.

Cliente PHP: “Reconstrucción y Presentación de Servicios Web de Aprendizaje”

El cliente PHP se encarga de la reconstrucción de los recursos empaquetados que han sido segmentados en un Servicio Web de Aprendizaje, para poder presentarlos de manera correcta. De igual manera, el cliente permite presentar los recursos de SWA tanto atómicos como compuestos en SOAP y REST. De acuerdo al mapa de navegación, las vistas corresponden a las siguientes funciones:

- **“index.php” (Vista):** es la ventana inicial del cliente, que presenta el título y los nombres de los autores.
- **“JAEM” (Módulo):** es el módulo que contiene al cliente antecedente “clientSoap” [Escobar Megchún, 2017].
 - **“test.php” (Vista):** es la ventana encargada de presentar al recurso contenido en un Servicio Web de Aprendizaje con el protocolo SOAP.
 - **“testRest.php” (Vista):** es la ventana encargada de presentar al recurso contenido en un Servicio Web de Aprendizaje con la arquitectura REST.
 - **“testComp.php” (Vista):** es la ventana encargada de presentar a los recursos contenidos en un Servicio Web de Aprendizaje Compuesto con el protocolo SOAP.

- **“IHFC” (Módulo):** es el módulo que contiene a la extensión implementada en este trabajo.
 - **“testBig.php” (Vista):** es la ventana encargada de reconstruir y presentar un recurso segmentado que se encuentra empaquetado en un SWA con el protocolo SOAP.
 - **“testRestBig.php” (Vista):** es la ventana encargada de reconstruir y presentar un recurso segmentado que se encuentra empaquetado en un SWA con la arquitectura REST.
 - **“testCompBig.php” (Vista):** es la ventana encargada de reconstruir y presentar múltiples recursos que han sido segmentados y se encuentran empaquetados en un Servicio Web de Aprendizaje Compuesto con el protocolo SOAP.
 - **“testCompRestBig.php” (Vista):** es la ventana encargada de reconstruir y presentar múltiples recursos que han sido segmentados y se encuentran empaquetados en un Servicio Web de Aprendizaje Compuesto con la arquitectura REST.

 Las interfaces desarrolladas en el Cliente PHP se detallan en la **Página 83**, correspondiente al **Anexo B**.

III. Ejecución de las Pruebas

Se definieron cuatro tipos de prueba diferentes, el primero consiste en verificar la correcta segmentación de recursos de aprendizaje y su generación como SWA. El segundo consiste en verificar la correcta reconstrucción de estos recursos contenidos para ser visualizados adecuadamente. El tercer tipo consiste en verificar la correcta composición de SWA teniendo recursos de pequeño y gran tamaño. Y el cuarto tipo consiste en reconstruir y presentar de manera adecuada los recursos contenidos en cada Servicio Web de Aprendizaje Compuesto. Para cada tipo de prueba se realiza la comparación en sus tiempos de respuesta entre SOAP y REST. El conjunto de estas pruebas se enlista en la Figura 5.8.

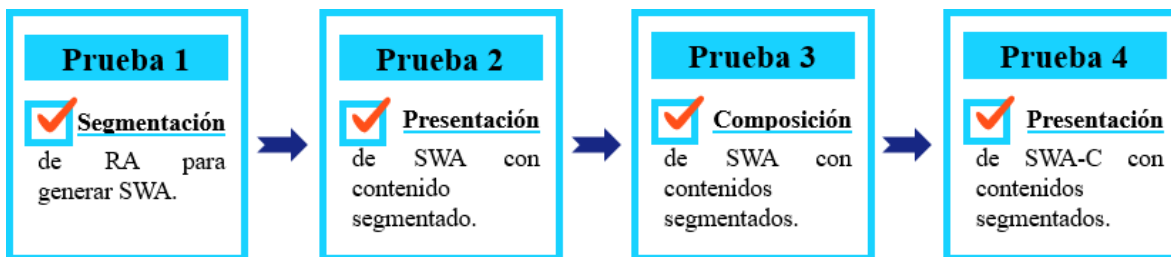


Figura 5.8. Lista de tipos de pruebas realizadas

Tipo de Prueba 1: “Segmentación de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje”

Esta prueba consiste en la segmentación de recursos de aprendizaje empaquetados en servicios web de tipos SOAP y REST, cuando un SWA es generado no siempre se le aplica

la actividad de segmentación, pero de ser aplicada se generan tantos segmentos en su cadena Base64 como sean necesarios que permitan reconstruir al recurso a su forma original.

De los 39 recursos de aprendizaje seleccionados en la Tabla 5.2, se generaron 78 Servicios Web de Aprendizaje, 39 SOAP y 39 REST. Teniendo un menor tiempo de respuesta con la arquitectura REST, tal como puede apreciarse en la Figura 5.9.

Comparación en los Tiempos de Segmentación

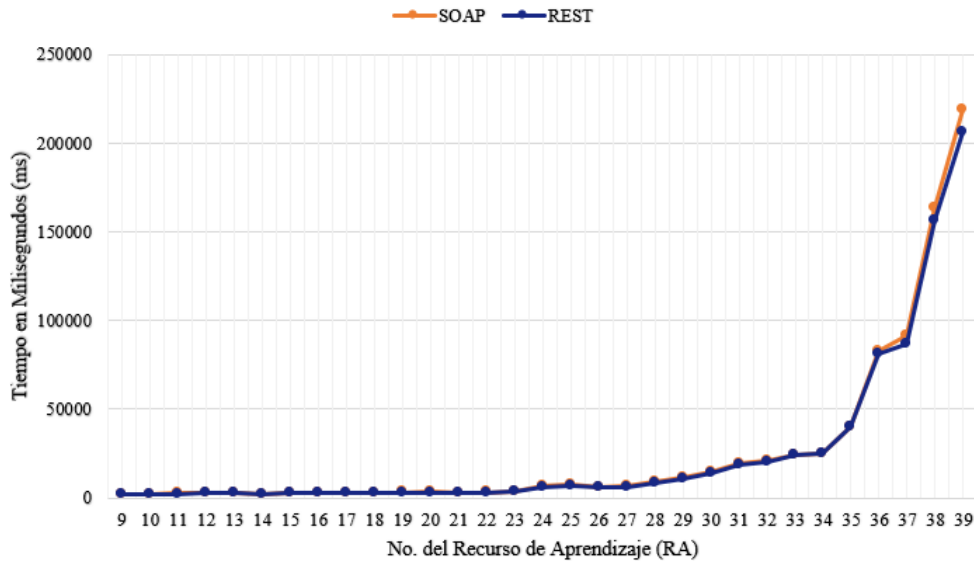


Figura 5.9. Comparación en los tiempos de segmentación de RA para generar SWA en SOAP y REST

La Figura 5.9 presenta los tiempos de segmentación de RA para la generación de SWA, donde la generación de SWA en SOAP toma mayor tiempo que la generación bajo REST. Aunque los resultados obtenidos demuestran un comportamiento similar de que a mayor tamaño del recurso y del servicio mayor será la cantidad de segmentos y el tiempo de respuesta, los tiempos pueden variar debido a las configuraciones del equipo y del entorno de pruebas. En esta gráfica se omiten los recursos del 1 al 8 ya que no fueron segmentados.

Tipo de Prueba 2: “Presentación de Servicios Web de Aprendizaje con contenido segmentado”

Esta prueba consiste en la reconstrucción de los recursos de aprendizaje que fueron segmentados en el Tipo de Prueba 1, los cuales fueron empaquetados como Servicios Web de Aprendizaje en SOAP y REST, con el propósito de poder presentarlos al usuario, sin que se vean afectados por su tamaño.

Se desarrolló un cliente PHP capaz de reconstruir Servicios Web de Aprendizaje con contenidos segmentados, uniendo parte a parte el recurso para poder presentarlo en un único producto. Donde de los 78 Servicios Web de Aprendizaje generados, todos fueron desplegados y presentados exitosamente. Para el proceso de reconstrucción se obtuvo un



El Tipo de Prueba 1 se detalla en la **Página 89**, que pertenece al **Anexo C**.



El Tipo de Prueba 2 se detalla en la **Página 92**, que pertenece al **Anexo C**.

menor tiempo de respuesta con los servicios de tipo REST, como se aprecia en la Figura 5.10. En la gráfica se omiten los servicios del 1 al 8 ya que sus recursos no fueron segmentados.

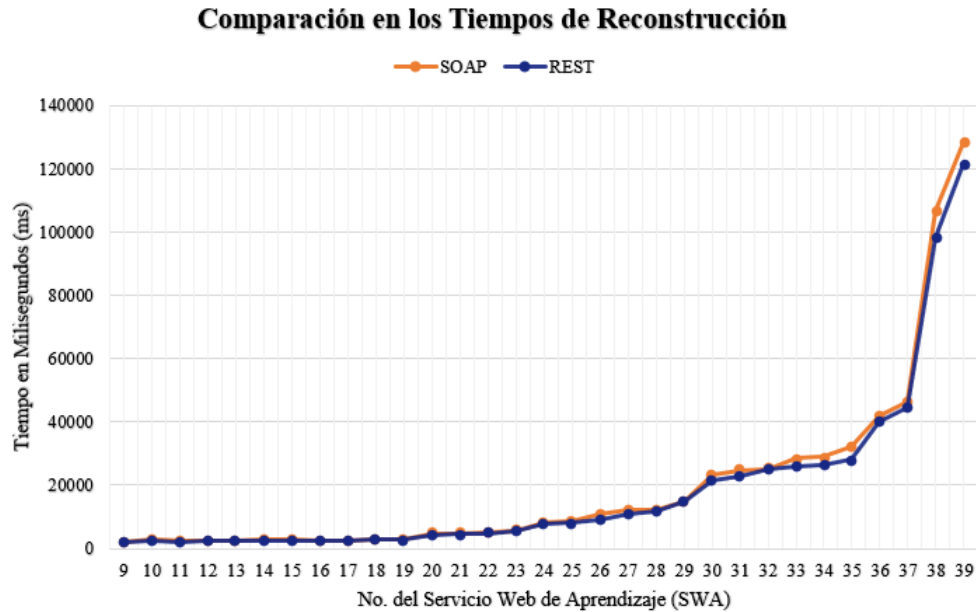


Figura 5.10. Comparación en los tiempos de reconstrucción de RA empaquetados como SWA

Un recurso segmentado y reconstruido correctamente puede presentarse en el cliente PHP para ser visualizado al usuario. Un ejemplo se muestra en la Figura 5.11 donde el encabezado muestra el nombre del recurso y debajo de éste, se despliega el SWA. Este mismo proceso ocurre para cada Servicio Web de Aprendizaje que desea visualizarse a través del cliente PHP, tanto para SOAP como REST.



Figura 5.11. Reconstrucción y presentación de un SWA

Tipo de Prueba 3: “Composición de Servicios Web de Aprendizaje con contenidos segmentados”

Esta prueba consiste en la composición de al menos dos Servicios Web de Aprendizaje generados mediante segmentación, tanto para SOAP como en REST. Donde se generaron Servicios Web de Aprendizaje Compuestos de manera exitosa con contenidos de pequeños y grandes tamaños.

Para propósitos de este trabajo, se generaron 8 SWA-C, siendo 4 para SOAP y 4 para REST, donde se empaquetaron los mismos recursos en distintos protocolos para verificar su correcta generación, teniendo recursos pequeños que no requirieron ser segmentados y recursos de gran tamaño que superaban el límite del almacenamiento dinámico en caso de no ser segmentados. Todos los Servicios Web de Aprendizaje Compuestos se generaron exitosamente teniendo recursos segmentados y no segmentados.



El Tipo de Prueba 3 se detalla en la **Página 95**, que pertenece al **Anexo C**.

Tipo de Prueba 4: “Presentación de Servicios Web de Aprendizaje Compuestos con contenidos segmentados”

Esta prueba consiste en la correcta presentación de los Servicios Web de Aprendizaje Compuestos que fueron generados en el Tipo de Prueba 3 para SOAP y REST, los cuales fueron presentados por el cliente PHP de manera correcta. Con lo que se muestra que se pueden realizar composiciones con recursos de diversos tamaños, donde se tomó como tiempo de respuesta a la presentación de todos los recursos ya visualizados por el navegador, teniendo a la arquitectura REST con la que presentan menores tiempos de respuesta. Esta comparación de SWA-C se aprecia en la Figura 5.12.



El Tipo de Prueba 4 se detalla en la **Página 97**, que pertenece al **Anexo C**.

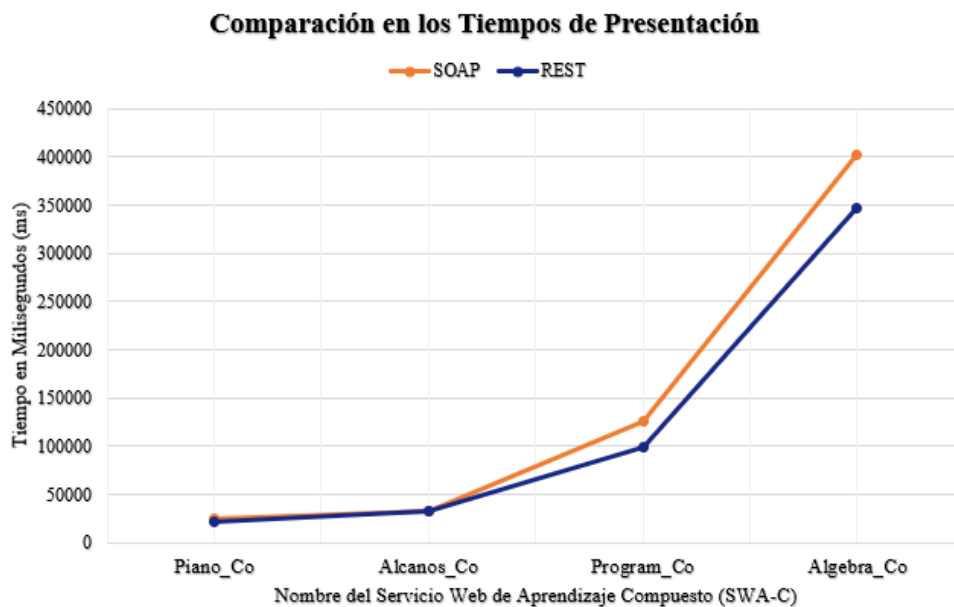


Figura 5.12. Comparación en los tiempos de presentación de SWA-C en SOAP y REST

Se identificó que a diferencia de cuando se trabaja con recursos menores a los 90 MB no existe problema en mezclar tipos de servicios SOAP con REST en un mismo servicio compuesto, pero cuando son recursos de mayor tamaño, es necesario generar servicios compuestos que sean del mismo tipo. Ocasionado por la sobrecarga de memoria del servidor. Por lo que se describen las siguientes condiciones al momento de realizar la composición:

- No se recomienda mezclar tipos de servicios, si el SWA-C contiene al menos un SWA mayor o igual a los 90 MB.
- Pueden mezclarse tipos de servicios, solo si todos los SWA a componer son menores a los 90 MB.
- Se recomienda mezclar SWA cuando el recurso contenido sea menor a los 10 MB. Con la finalidad de tener un menor tiempo de respuesta para la presentación de sus recursos.

La Figura 5.13 muestra un ejemplo de SWA-C en donde se visualizan todos los SWA atómicos que participan en la composición.

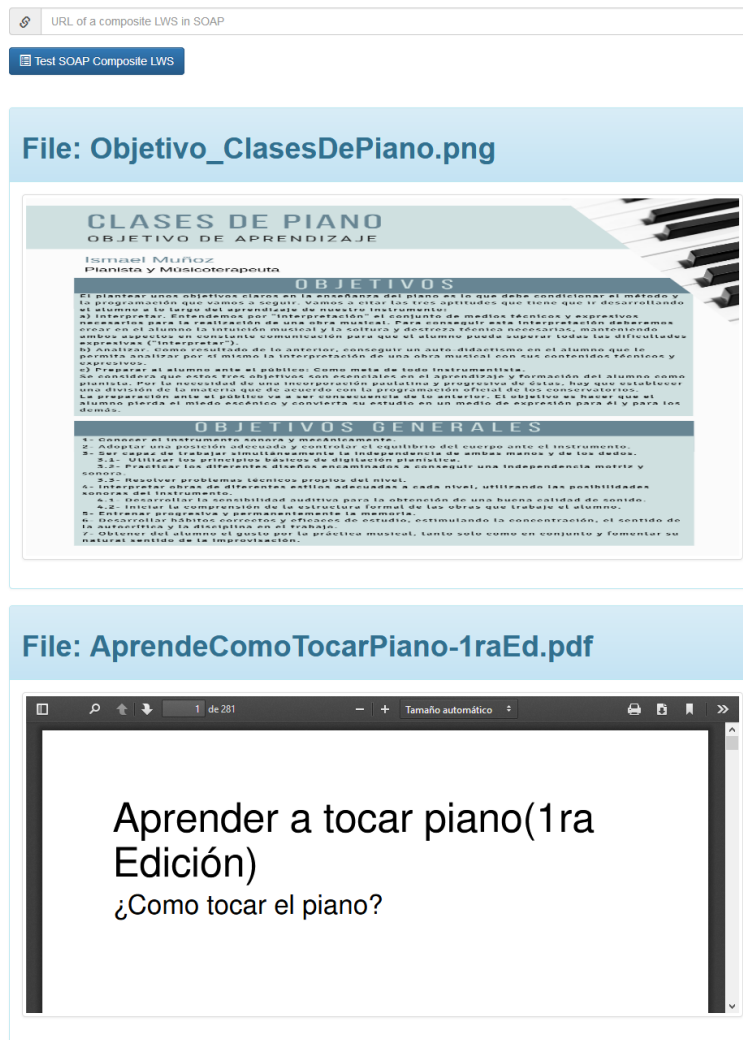


Figura 5.13. Presentación exitosa de un Servicio Web de Aprendizaje Compuesto

5.4. Resultados y Conclusiones de las Pruebas

Los resultados de las pruebas realizadas proporcionan datos importantes sobre la estrategia de gestión implementada en este trabajo de investigación y que considera las actividades de segmentación, reconstrucción, generación, composición y presentación de los recursos de aprendizaje contenidos en los Servicios Web de Aprendizaje, las cuales se describen en esta sección.

Se cumplió con los cuatro objetivos de las pruebas descritos en la [Sección 5.2.1](#), los cuales son detallados a continuación:

- En la Prueba Preliminar 2 de generación de Servicios Web de Aprendizaje para los casos de pruebas mediante la estrategia de gestión, la generación se realiza de manera correcta y sin presentar daños en sus cadenas de datos en Base64. Con lo que se verifica el primer objetivo de las pruebas, donde todos los SWA generados cumplirán con la actividad de segmentación de ser necesario y posteriormente con la actividad de reconstrucción para el ordenamiento de sus segmentos dentro del WSDL para servicios de tipo SOAP, y añadiendo métodos de implementación que conforman al SWA en el caso de REST.
- El Tipo de Prueba 1 consiste en generar SWA de tipo SOAP y REST, los cuales sirven para ejecutar el Tipo de Prueba 2. Cumpliendo de este modo el segundo objetivo de pruebas de visualizar Recursos de Aprendizaje que han sido segmentados y se encuentran contenidos en un SWA.
- De igual manera, se verificó la correcta ejecución y presentación para visualizar múltiples Recursos de Aprendizaje que han sido segmentados y reconstruidos, y se encuentran empaquetados como un Servicio Web de Aprendizaje Compuesto. Con lo que se cumplió con el tercer objetivo de las pruebas de visualizar múltiples Recursos de Aprendizaje que han sido segmentados y se encuentran contenidos en un SWA-C.
- El cuarto y último objetivo de las pruebas, consistió en comparar los tiempos de respuesta en la entrega de los recursos contenidos en los Servicios Web de Aprendizaje tanto para servicios de tipo SOAP como para servicios de tipo REST. Este se cumple con los resultados obtenidos en el Tipo de Prueba 2 para SWA atómicos y del Tipo de Prueba 4 para SWA compuestos. Donde en ambos se obtuvieron menores tiempos de respuesta con los servicios de tipo REST.

Adicionalmente, se describen ciertos comportamientos que se identificaron durante el desarrollo del trabajo y la realización de las pruebas:

La actividad de segmentación se cumple cuando el recurso de aprendizaje es igual o mayor a los 9,000,000 Bytes (8.58 MB), cuyos segmentos resultantes son de 3,000,000 Bytes (2.86 MB) cada uno, con excepción del último que podría ser menor, y no presentan pérdida de información ni daños en su estructura. Estas medidas fueron establecidas a partir de la comparación de 8 rangos de tamaños descritos en la Prueba Preliminar 1, donde se descartaron a los rangos 1, 2, 4 y 6 por generar código blanco, el cual dañaba al contenido

del recurso. Y se seleccionó al rango 3 por presentar menor tiempo de respuesta para la reconstrucción del recurso sin presentar daños en su contenido.

Uno de los motivos por el cual se definió que uno de los límites a probar sea de 8.58 MB para realizar la segmentación, es que al visualizar un recurso de aprendizaje que sea menor a los 10 MB, presentaba un tiempo de respuesta no mayor a 10 segundos. Cuando se trabaja con recursos de mayor tamaño los tiempos van incrementando considerablemente. En cambio, cuando se trabaja con recursos de 9 MB o de menor tamaño, el lapso de tiempo para ser presentados es inferior a los 10 segundos.

Se identificó que, aunque el tamaño de los segmentos obtenidos al realizar la actividad de segmentación puede ser cualquier tamaño en Bytes divisible entre tres, no es recomendable definir un tamaño muy bajo debido a que se generarán demasiadas clases que saturarán el límite de clases que puede desplegar el servidor. Esto debido a la estrategia implementada en este trabajo, donde se generan tantas clases como segmentos sean necesarios. Por ejemplo, si se define un tamaño máximo para cada segmento de 30 KB en un recurso con 200 MB se generarán más de 6,000 clases las cuales no podrán ser desplegadas por el servidor y por tal motivo no se visualizará el recurso.

La reconstrucción de los Servicios Web de Aprendizaje con contenido segmentado ocurrió de manera exitosa con la estrategia de gestión propuesta, en el caso de SOAP consiste en la modificación de su documento WSDL, donde se le añade la etiqueta “Partes” que hace referencia al número de segmento, el cual es redireccionado a una clase que lleva el mismo nombre y contiene un fragmento de la cadena de Base64 del recurso. Lo que permite su reconstrucción de manera correcta. Para el caso de REST, la reconstrucción ocurre a través de los métodos de implementación que conforman al SWA. El cliente PHP cuenta con un controlador para servicios atómicos y otro para servicios compuestos, encargados de recibir a los segmentos que conforman el recurso y presentarlos en un único producto.

La estrategia de gestión solventa la problemática en el límite del almacenamiento dinámico de la memoria de asignada en la Máquina Virtual de Java, que no permite visualizar recursos de aprendizaje mayores a los 50 MB debido a un desbordamiento. Por lo que se aporta un método que permite entregar Recursos de Aprendizaje de gran tamaño a través de servicios web con los tipos SOAP y/o REST.

Actualmente, los sistemas gestores de aprendizaje proveen cursos o materiales educativos individuales para que los usuarios finales puedan adquirir nuevos conocimientos, estos ofrecen recursos de aprendizaje menores a los 25 MB, esto debido a que esta es una medida estándar en la configuración de los servidores para la carga y presentación de recursos. Sin embargo, la estrategia de gestión desarrollada es una aportación para que estas plataformas, o incluso otras cuyo enfoque no sea la educación, puedan aumentar estas limitantes en el tamaño de los recursos. También hay que considerar, que los recursos de aprendizaje de gran tamaño suelen ser vídeos de larga duración y/o de alta definición, o documentos con mucha carga de información, por lo que no es muy común encontrarlos.

En esta estrategia se identifican ciertas características que permiten la segmentación y la reconstrucción automática de Recursos de Aprendizaje contenidos en Servicios Web de Aprendizaje, las cuales son el límite para realizar la segmentación, el tamaño de los segmentos del recurso, el método de reconstrucción y validar el contenido de los segmentos. Aunque los valores utilizados para este trabajo de tesis fueron satisfactorios, estos podrían modificarse o se podrían implementar nuevos criterios para analizar y comparar sus comportamientos.

Un problema que persiste, es que la herramienta Java permite generar Servicios Web de Aprendizaje de cualquier tipo de recurso, pero un cliente PHP no es capaz de visualizarlos todos, esto debido a que los navegadores solo pueden presentar recursos de tipo Extensión Multipropósito de Correo de Internet (MIME). Por lo que al generar SWA de otros formatos como los documentos de la paquetería de Microsoft Office 2013 o versiones superiores, cuyas extensiones son DOCX, PPTX y XLSX no podrán ser visualizadas, y se enviarán como archivo a descargar en el equipo del cliente. Esto podría solucionarse con una extensión o un plugin que permita al navegador visualizar dichas extensiones, o con la implementación de un método que permita visualizar dichos formatos de recursos.

Se utilizó el entorno de Apache JMeter ya que permite aislar al equipo de procesos que interfieren con la realización de las pruebas, aunque los tiempos de respuesta variaron debido al uso de otras funciones realizadas por el sistema operativo. Los resultados obtenidos mediante este entorno, para los procesos de segmentación, reconstrucción y composición de los Servicios Web de Aprendizaje indican que la arquitectura REST presenta menores tiempos de respuesta en todos los casos establecidos. Para los propósitos de pruebas en este trabajo se utilizaron los parámetros definidos en JMeter de 10 hilos (usuarios) con un lapso de tiempo de 5 segundos, el cual permitía obtener un tiempo de respuesta más preciso a los procesos en cada prueba. Sin embargo, se debe considerar que se pueden obtener resultados diferentes si se definen distintos valores para los parámetros en JMeter. También es necesario reiniciar el equipo para iniciar el servidor desde cero, debido a que existe cierta permanencia en el uso de la memoria cache, lo que puede ocasionar ruido al momento de obtener los tiempos de respuesta.

Aunque el entorno de pruebas permitió aislar diversos procesos del equipo, al realizar ciertas cantidades de pruebas quedaba un pequeño espacio ocupado en la memoria caché. Por lo que debía reiniciarse el equipo para que el servidor se iniciará desde cero, y de este modo se obtuvieran mejores tiempos de respuesta al realizar las pruebas nuevamente.

Pese a que los tiempos de respuesta fueron menores en REST para todos los casos de prueba establecidos, se identifica que no hay una diferencia significativa entre SOAP y REST, ya que los tiempos en milisegundos varían en proporciones mínimas y no afectan al uso ni la interacción del usuario.

Ocurre un comportamiento esperado en los resultados obtenidos, donde ha mayor tamaño del Recurso de Aprendizaje, entonces mayor será el tiempo para su segmentación, reconstrucción y presentación.

Capítulo

6

Conclusiones & Trabajo Futuro

En este capítulo se presentan las conclusiones de acuerdo a los resultados obtenidos, así como las observaciones que se identificaron en este trabajo de investigación y un análisis de los trabajos futuros.

6.1. Conclusiones

Como resultado de las pruebas presentadas, se concluye que este trabajo cumplió con el objetivo general y sus objetivos específicos, ya que la implementación de la estrategia de gestión aplicada a Recursos de Aprendizaje para generar Servicios Web de Aprendizaje con contenidos de gran tamaño permitió presentarlos exitosamente, además de permitir la composición entre estos, de manera que múltiples servicios web puedan ser utilizados en una composición de servicios sin verse afectados por su tamaño.

6.1.1. Objetivos Cumplidos

Los objetivos propuestos en este trabajo de investigación y las actividades realizadas para cumplirlos se detallan en la Tabla 6.1.

Tabla 6.1. Objetivos cumplidos

Objetivo	Comentarios
Definir una estrategia de gestión de Recursos de Aprendizaje para que puedan participar en procesos de generación de Servicios Web de Aprendizaje con contenidos de gran tamaño.	Se definió una estrategia de gestión que permite presentar de manera correcta a los <i>Servicios Web de Aprendizaje</i> con contenidos de gran tamaño, la cual consiste de dos actividades, la primera para la segmentación de <i>Recursos de Aprendizaje</i> que se realiza correctamente en el Tipo de Prueba 1 , y la segunda actividad se encarga de establecer el medio para reconstruir a estos recursos empaquetados como <i>Servicios Web de Aprendizaje</i> , tanto en SOAP como en REST, los cuales se presentaron de manera correcta en el Tipo de Prueba 2 .
Conocer las características para una guía de reconstrucción automática.	Se identificaron las características para la segmentación y reconstrucción automática de <i>Recursos de Aprendizaje</i> contenidos en <i>Servicios Web de Aprendizaje</i> , las cuales son: el límite para realizar la segmentación, el tamaño asignado para los segmentos del recurso, el método de reconstrucción y la validación del contenido de los segmentos, mismas que se encuentran descritas en el Capítulo 4 .
Realizar la composición de servicios web cuyos contenidos fueron segmentados para gestionar su contenido y presentarlo de manera correcta.	En el Tipo de Prueba 3 se realizaron las composiciones de <i>Servicios Web de Aprendizaje</i> con contenidos de pequeños y grandes tamaños que fueron segmentados en SOAP y REST, y en el Tipo de Prueba 4 se hizo la reconstrucción y presentación de los recursos contenidos en los <i>Servicios Web de Aprendizaje Compuestos</i> donde se entregaron de manera correcta.

6.1.2. Conclusiones Específicas

Generación de Servicios Web de Aprendizaje

- Persiste el hecho que, aunque haya herramientas que permitan generar Servicios Web de Aprendizaje con cualquier tipo de recurso, el cliente no es capaz de visualizarlos todos debido a que los navegadores solo pueden visualizar recursos de tipo MIME. Esto podría solucionarse con una extensión o un plugin que permita al navegador visualizar dichas extensiones, pero cuyos formatos no se encuentran dentro de los alcances de este trabajo.

- Se logró presentar recursos empaquetados en servicios web de aprendizaje atómicos de hasta 688 MB, superando el límite de almacenamiento dinámico (Java Heap Space) de 50 MB. Aunque los Servicios Web de Aprendizaje con contenido podrían ser de un tamaño mayor.
- La generación y presentación de Servicios Web de Aprendizaje con contenidos de pequeño o de gran tamaño, se da de manera exitosa mediante la estrategia de gestión, tanto para servicios de tipo SOAP como para tipo REST.
- Los Recursos de Aprendizaje al ser codificados y empaquetados en Servicios Web de Aprendizaje aumentan su tamaño original entre 29% y 30% aproximadamente, por lo que al trabajar con contenidos de gran tamaño se recomienda alojarlos en distintos proyectos separando a cada tipo de servicio, con el propósito de evitar saturar la memoria del servidor.
- Para los casos de prueba establecidos, los servicios de tipo REST tuvieron tiempos de respuesta menores para la generación de servicios web de aprendizaje y la actividad de segmentación, así como para la presentación por medio de la actividad de reconstrucción. Sin embargo, estos tiempos podrían variar dependiendo de las configuraciones del equipo y del entorno de pruebas, así como otras funciones que realiza el sistema operativo y no pueden ser detenidas.
- A pesar de que las diferencias entre los tiempos de respuesta para SOAP y REST no son muy significativos debido a que son cantidades pequeñas en milisegundos, se recomienda generar Servicios Web de Aprendizaje de tipo REST para agilizar la entrega de los recursos. Aunque con SOAP existe una mayor protección del contenido al generar capas de etiquetado, se debe considerar que se trabaja con recursos de aprendizaje los cuales se encuentran abiertos a todos los usuarios para que puedan adquirir nuevos conocimientos, por lo que no son tan propensos a posibles ataques de robo de información.

Composición de Servicios Web de Aprendizaje

- Con la composición de Servicios Web de Aprendizaje se pueden unir dos o más servicios con contenidos de pequeño y/o gran tamaño, permitiendo de esta manera poder presentar múltiples Recursos de Aprendizaje sin que se vean afectados por sus tamaños.
- La herramienta Java permite la composición de servicios compuestos en SOAP, en REST o en una mezcla de ambos. Sin embargo, al trabajar con recursos de gran tamaño se recomienda utilizar un servicio compuesto en un único tipo de servicio y cuyas invocaciones sean del mismo también. Ya que, al desplegar recursos mayores a los 90 MB en un mismo Servicio Web de Aprendizaje Compuesto, pero con distintos tipos de servicios, es posible que estos servicios no puedan ser ejecutados debido a la sobrecarga de memoria del servidor, por el uso de las librerías de ambos protocolos trabajando simultáneamente. Estas condiciones se describen en el **Tipo de Prueba 4**.

- El cliente PHP tiene un estándar definido de 1 GB para cargar y presentar recursos contenidos en Servicios Web de Aprendizaje, el cual puede modificarse desde el código. Esta es una medida de prueba para simular la visualización de recursos de gran tamaño en una plataforma educativa.

Estrategia de Gestión para Recursos de Aprendizaje

- La metodología de solución propuesta en este trabajo pudo ser agregada como una extensión a la herramienta [Escobar Megchún, 2017] denominada “LWS”, ya que esta se encuentra basada en el patrón de diseño Strategy. Por lo que se modificaron las clases ya existentes para adaptar al comportamiento que requiera el usuario y se agregaron las clases correspondientes a la estrategia de gestión definida.
- La estrategia de gestión definida permitió la correcta segmentación, reconstrucción y presentación de recursos de gran tamaño contenidos en un Servicio Web de Aprendizaje. Así como su correcta presentación en Servicios Web de Aprendizaje Compuestos.
- Adicionalmente, se analizaron los comportamientos en distintos formatos de recursos como MP4, PDF, DOC y PPT. Siendo el comportamiento similar de que ha mayor tamaño del recurso, mayor será la cantidad de los segmentos y de los tiempos de respuesta. Donde para los casos de prueba establecidos, los servicios de tipo REST tuvieron tiempos de respuesta menores y que no existe alguna alteración en los tipos de formatos de los recursos, los cuales se detallan en el [Anexo D](#).

6.2. Aportaciones

Las principales aportaciones realizadas con este trabajo de investigación son:

- La estrategia de gestión que permite presentar de manera correcta a los Servicios Web de Aprendizaje con contenidos de gran tamaño, la cual consiste de dos actividades:
 - La actividad de segmentación donde se identifican las características necesarias para segmentar un recurso de gran tamaño sin dañar su contenido, como son: el límite para realizar la segmentación, el tamaño máximo que tendrá cada segmento y la validación de su contenido para evitar la pérdida de información o el daño en su cadena de datos.
 - La actividad de reconstrucción donde se modifica el documento WSDL para ordenar los segmentos en el caso de SOAP, o la implementación de métodos en la interfaz del SWA capaz de conservar la secuencia de los segmentos para el caso de REST.

6.3. Trabajos Futuros

En este trabajo se da una alternativa para resolver la problemática del límite de almacenamiento dinámico en recursos contenidos de gran tamaño que se encuentran empaquetados en servicios web de aprendizaje. Los trabajos futuros contemplados se describen a continuación:

- Realizar los procesos de segmentación, reconstrucción, generación, presentación y composición de Servicios Web de Aprendizaje con otros protocolos o herramientas para la transferencia de datos, como GraphQL.
- Definir otra estrategia de gestión para los procesos de segmentación y reconstrucción de los Recursos de Aprendizaje, como la transmisión por secuencias de *streaming* que considera el ancho de banda y tiempos de latencia entre el cliente – servidor para tener una presentación de los recursos en tiempo real.
- Implementar un método o estrategia que permita amplificar el tamaño en el límite de carga de los servicios desplegados, para que sean presentados de manera correcta sin realizar segmentación.
- Aplicar y evaluar con pruebas de similitud a los recursos antes de ser segmentados y posteriormente a su reconstrucción y presentación, con el propósito de analizar que el contenido del recurso no se encuentre dañado ni con pérdidas de información.
- Realizar pruebas para verificar los comportamientos en los procesos de segmentación, reconstrucción, presentación y composición de Servicios Web de Aprendizaje en la memoria caché.
- Se recomienda realizar pruebas de segmentación y reconstrucción considerando las características del contenido de los recursos, ya que este trabajo considera solamente el tamaño, tipo de servicio y los formatos.
- Aplicar la estrategia de gestión definida en este trabajo a otras tecnologías, como podría ser el caso de los recursos que son enviados a través de mensajes de correo electrónico.

Referencias

- Alfonso, M. I. (2012). *Servicios Web y SOA*. Alicante, España: pp. 4-15. Universidad de Alicante. Dpto. de Ciencia de la Computación e Inteligencia Artificial.
- Apache. (2019). *Apache JMeter™ version 5.1.1 API*. Obtenido de The Apache Software Foundation: <https://JMeter.Apache.org>
- Barakat, L., Miles, S., & Luck, M. (2018). Adaptive composition in dynamic service environments. *Elsevier. Future Generation Computer Systems*, 215-228. DOI: 10.1016/j.future.2016.12.003.
- Begam, M. F., & Ganapathy, G. (2014). Ontology Based Dynamic e-Learning Flow Composition of Learning Web Services. *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, núm. 22, 4723-4735. DOI: 10.19026/rjaset.7.858.
- Belmabrouk, K., Bendella, F., & Bouzid, M. (2016). Multi-Agent Based Model for Web Service Composition. *Article Published in International Journal of Advanced Computer Science and Applications(IJACSA), Volume 7, Issue 3*, 144-150. DOI: 10.14569/IJACSA.2016.070320.
- Benítez Domínguez, S. (2018). *Evaluación de capacidades SOAP y REST para entregar Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Bentahar, J., Yahyaoui, H., Kova, M., & Maamar, Z. (2013). Symbolic model checking composite Web services using operational and control behaviors. *Expert Systems with Applications. Vol. 40, Issue 2*, 508-522. DOI: 10.1016/j.eswa.2012.07.069.
- Cerami, E. (2002). *Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL (1st Edition)*. CA, United States of America: O'Reilly Media.
- Contreras Sierra, E. R. (2013). El concepto de estrategia como fundamento de la planeación estratégica. *Pensamiento & Gestión, no. 35*, 152-181. ISSN: 1657-6276.
- Damavandi, T. R., Jayarathna, S., & Sun, Y. (2018). Modeling CDC Data for Real Time Feature Estimation. *IEEE International Conference on Information Reuse and Integration (IRI)*, 248-251. DOI: 10.1109/IRI.2018.00045.
- Delgado Fernández, P. (2012). *Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Drucker, P. (2007). *The practice of management. The classic Drucker collection (Edición revisada, reimpresión)*. M. Belbin (Ed.) USA: Elsevier.
- Erl, T. (2005). *Service-Oriented Architecture. Concepts, Technology and Design*. Crawfordsville, Indiana: Prentice Hall. pp. 3-6.
- Escobar Megchún, J. A. (2017). *Generador de Servicios Web de Aprendizaje Compuestos a partir de Recursos Educativos*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Espinosa Pérez, P. (2016). *Esquema de clasificación de Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Fernández Redondo, M. (2006). *Sistemas Operativos. Administración de Memoria*. Castellón, España: Universidad Jaime I. Departamento de Ingeniería y Ciencia de los Computadores.

- Friesen, N. (2005). Interoperability and Learning Objects: An Overview of E-Learning Standardization. *Interdisciplinary Journal of Knowledge and Learning Objects*. Vol. 1, 23-31. DOI: 10.28945/408.
- Gabrel, V., Manouvrier, M., & Murat, C. (2015). Web services composition: Complexity and models. *Elsevier. Discrete Applied Mathematics*, 100-114. DOI: 10.1016/j.dam.2014.10.020.
- Golab, L., & Ozsu, M. T. (2010). *Data Stream Management*. Synthesis Lectures on Data Management. University of Waterloo: Morgan & Claypool. DOI: 10.2200/S00284ED1V01Y201006DTM005.
- Gómez, F. S., & Solarte, M. F. (2014). A survey of web-resource applications, within learning management system. LRN, implemented as e-learning services. *Tecnura Vol. 18, Núm. 39*, 165-180. DOI: 10.14483/issn.2248-7638.
- Gregorio, J. d. (2017). *Sistema Constructor de Recursos Educativos como Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Gutiérrez Juárez, G. (2018). *Sistema integrador de componentes de gestión de Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Jing, J., Helal, A., & Elmagarmid, A. (1999). Client-Server Computing in Mobile Environments. *ACM Computing Surveys*. Vol. 31, Issue 2, 117. DOI: 10.1145/319806.319814.
- Kan, Q., Yang, Y., Zhen, S., & Liu, W. (2010). A Unified Record Linkage Strategy for Web Service Data. *Third International Conference on Knowledge Discovery and Data Mining*, 253-256. DOI: 10.1109/wkdd.2010.134.
- Kolesnikov, A. M., Lomachenko, T. I., Kokodey, T. A., Khitushchenko, V. V., & Mihailov, Y. I. (2019). A Strategy of Visualization and Interactive Support for University Level Educational Digitalization. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, DOI: 10.1109/EIConRus.2019.8657211.
- Krouska, A., Troussas, C., & Virvou, M. (2017). Comparing LMS and CMS Platforms Supporting Social e-Learning in Higher Education. *IEEE*, 1-6. DOI: 10.1109/IISA.2017.8316408.
- Kumar, N., Malik, P., Bhardwaj, S., & Jain, S. C. (2016). Comparative analysis of deduplication techniques for enhancing storage space. *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 480-487. DOI: 10.1109/PDGC.2016.7913243.
- Liu, L., Zhou, C., Zhang, X., Guo, Z., & Li, C. (2014). Probabilistic chunk scheduling approach in parallel multiple-server DASH. *IEEE Visual Communications and Image Processing Conference*, 5-8. DOI: 10.1109/VCIP.2014.7051490.
- Liu, X., Liao, J., & Shen, Q. (2013). Adaptive CDCLZW algorithm for data compression. *5th IEEE International Conference on Broadband Network & Multimedia Technology*, 80-84. DOI: 10.1109/ICBNMT.2013.6823919.
- LOMWG12 - Working Group for Learning Object Metadata. (2009). IEEE Standard for Learning Object Metadata. *IEEE Std 1484.12.1-2002*. DOI: 10.1109/IEEESTD.2002.94128
- López Orduña, E. A. (2013). *Definición de elementos del WSDL para Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Madjarov, I., & Boucelma, O. (2006). Data and application integration in learning content management systems: a web services approach. *Proceedings of the First European conference on Technology*

- Enhanced Learning: innovative Approaches for Learning and Knowledge Sharing* (págs. 272-286). Marseille: Aix-Marseille Universités. DOI: 10.1007/11876663_22.
- Malki, A., Barhamgi, M., Benslimane, S.-M., Benslimane, D., & Malki, M. (2015). Composing Data Services with Uncertain Semantics. *IEEE Transactions on Knowledge and Data Engineering. Issue Vol. 6, No. 1*, 1-14. DOI: 10.1109/TKDE.2014.2359661.
- MDN Web Docs. (2013). *Base64: Codificando y decodificando*. Obtenido de https://developer.mozilla.org/es/docs/Web/API/WindowBase64/Base64_codificando_y_decodificando
- MDN Web Docs. (2019). *HTTP: Tipos MIME*. Obtenido de https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_tipos
- Milanovic, N., & Malek, M. (2004). Current Solutions for Web Service Composition. *IEEE Internet Computing, vol. 8, issue 6*, 51-59. DOI: 10.1109/MIC.2004.58.
- Morgado, L., Almeida, Á., Vilela, A., Pires, B., Paredes, H., Fonseca, B., . . . Santos, A. (2016). Integrating Virtual Worlds with Learning Management Systems: The MULTIS Approach. *15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, 167-172. DOI: 10.1109/IUCC-CSS.2016.031.
- Muła, W., & Lemire, D. (2018). Faster Base64 Encoding and Decoding Using AVX2 Instructions. *ACM Transactions on the Web Vol. 12, No. 3, Article 20*, 1-26. DOI: 10.1145/3132709.
- Nakano, Y., Yamato, Y., Takemoto, M., & Sunaga, H. (2007). Method of creating web services from web applications. *IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07)*, 1-7. DOI: 10.1109/SOCA.2007.28.
- Navarro Luna, J. (2001). Administración de Oracle: Parte I. *Grupo Cartuja Informática. Centro de Formación del Profesorado de Sevilla*, 1-142.
- Nishida, H., & Nguyen, T. (2013). Optimal Client-Server Assignment for Internet Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems. Vol. 24, Issue 3*, 565-575. DOI: 10.1109/TPDS.2012.169.
- Omer, A. M., & Schill, A. (2011). Automatic management of cyclic dependency among web services. *14th IEEE International Conference on Computational Science and Engineering*, 44-51. DOI: 10.1109/CSE.2011.22.
- Oracle Docs. (2016). *Tuning Java Virtual Machines (JVMs)*. Obtenido de https://docs.oracle.com/cd/E15523_01/web.1111/e13814/jvm_tuning.htm#PERFM150
- Padmavathi, G., Muthukumar, M., & Thakur, S. K. (2010). Implementation and Comparison of Different Segmentation Algorithms used for Underwater Images Based on Nonlinear Objective Assessments. *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 393-397. DOI: 10.1109/ICACTE.2010.5579301.
- Parejo, J. A., Segura, S., Fernandez, P., & Ruiz-Cortés, A. (2014). QoS-aware web services composition using GRASP with Path Relinking. *Expert Systems with Applications 41*, 4211-4223. DOI: 10.1016/j.eswa.2013.12.036.
- Partheeban, N., & SankarRam, N. (2014). e-Learning Management System Using Web Services. *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 1-7. DOI: 10.1109/ICICES.2014.7033900.

- Potti, P. K., Ahuja, S., Umapathy, K., & Prodanoff, Z. (2012). Comparing Performance of Web Service Interaction Styles: SOAP vs. REST. *Proceedings of the Conference on Information Systems Applied Research*, 1-24. ISSN: 2167-1508.
- Rabahallah, K., & Ahmed-Ouamer, R. (2015). Creating e-Learning Web Services Towards Reusability of functionalities in creating e-learning systems. *Global Summit on Computer & Information Technology (GSCIT)*, 1-6. DOI: 10.1109/GSCIT.2015.7353329.
- Rai, G. N., & Gangadharan, G. R. (2015). Set partition and trace based verification of Web service composition. *6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)*, Procedia Computer Science 52. 278-285. DOI: 10.1016/j.procs.2015.05.081.
- Rebolledo Castañeda, L. C. (2017). *Análisis y Definición de una Métrica para evaluar la Legibilidad en Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Salinas Roman, H. (2017). *Modelado Orientado a Objetos para Medir la Calidad en Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Seceleanu, T., Leppanen, V., Suomi, J., & Nevalainen, O. (2005). Resource allocation methodology for the segmented bus platform. *Proceedings 2005 IEEE International SOC Conference*, 129-132. DOI: 10.1109/SOCC.2005.1554479.
- Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences*, vol. 280, 218-238. DOI: 10.1016/j.ins.2014.04.054.
- Sicilia, M.-A. (2016). Reusabilidad y reutilización de objetos didácticos: mitos, realidades y posibilidades. *RED. Revista de Educación a Distancia*. Núm. 50. Art. 6, 1-8. DOI: 10.6018/red/50/6.
- Sreenath, V., & George, B. (2015). An Easy-to-Interface CDC With an Efficient Automatic Calibration. *IEEE Transactions on Instrumentation and Measurement*. Vol. 65, Issue 5, 960-967. DOI: 10.1109/TIM.2015.2495719.
- Szabo, M., & Flesher, K. (2002). CMI Theory and Practice: Historical Roots of Learning Management Systems. *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002* (págs. 926-936. ISBN: 978-1-880094-46-4). Montreal, Canada: Association for the Advancement of Computing in Education (AACE).
- Uriostegui Cuadra, M. (2019). *Definición e implementación de restricciones para un esquema de clasificación de Servicios Web de Aprendizaje*. Cuernavaca, Morelos, México: Tesis de Maestría en Ciencias, CENIDET.
- Valbuena, S. J., & Mora, V. C. (2013). Comparativo entre los sistemas de gestión de aprendizaje Moodle y Maat Gknowledge. *Inge Cuc*, vol. 9, no. 1, 183-195. ISSN: 0122-6517.
- Valenzuela Robles, B. D. (2017). *Integración de Recursos de Aprendizaje en Moodle con base en el Modelo de Servicios Web*. Cuernavaca, Morelos, México: Tesis de Doctorado en Ciencias, CENIDET.
- Valenzuela, B. D., Fragoso, O. G., Santaolaya, R., & Muñoz, J. (2017). Educational Resources as Learning Web Services, an Alternative Point of View to Learning Objects. *IEEE Latin America Transactions*, vol. 15, Issue 4, 711-719. DOI: 10.1109/TLA.2017.7896399.

- Valenzuela, B. D., Fragoso, O. G., Santaolaya, R., Muñoz, J., & Rojas, J. C. (2014). Integration of learning Web services into learning management systems. *IEEE Central America and Panama Convention (CONCAPAN XXXIV)*, 1-6. DOI: 10.1109/CONCAPAN.2014.7000409.
- Van Assche, F. J., & Vuorikari, R. (2006). A framework for quality of learning resources. *Handbook on Quality and Standardisation in E-Learning*, 443-456. DOI: 10.1007/3-540-32788-6_29.
- W3C Working Group. (11 de Febrero de 2004). *Web Services Architecture*. Obtenido de World Wide Web Consortium (W3C): <https://www.w3.org/TR/ws-arch/>
- Wang, H., Chen, X., Wu, Q., Yu, Q., Hu, X., Zheng, Z., & Bouguettaya, A. (2017). Integrating Reinforcement Learning with Multi-Agent Techniques for Adaptive Service Composition. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*. Vol. 12, Issue 2, 1-42. DOI: 10.1145/3058592.
- Wang, H., Wang, X., Zhang, X., Yu, Q., & Hu, X. (2015). Effective service composition using multi-agent reinforcement learning. *Knowledge-Based Systems*, vol. 92, issue C, 151-168. DOI: 10.1016/j.knosys.2015.10.022.
- Won, Y., Lim, K., & Min, J. (2015). MUCH: Multithreaded Content-Based File Chunking. *IEEE Transactions on Computers*. Vol. 64, Issue 5, 1375-1388. DOI: 10.1109/TC.2014.2322600.
- Xu, J., Song, L., & Xie, R. (2017). Two-Stream Deep Encoder-Decoder Architecture for Fully Automatic Video Object Segmentation. *IEEE Visual Communications and Image Processing (VCIP)*, 1-4. DOI: 10.1109/VCIP.2017.8305089.
- Zaychenko, S., Leshtaev, P., Gureev, B., & Shliakhtun, M. (2016). Structural CDC analysis methods. *IEEE East-West Design & Test Symposium (EWDTS)*, 1-4. DOI: 10.1109/EWDTS.2016.7807671.
- Zeng, L., Ngu, A. H., Benatallah, B., Podorozhny, R., & Lei, H. (2008). Dynamic composition and optimization of Web services. *Distributed and Parallel Databases*. Vol. 24, Issue 1-3, 45-72. DOI: 10.1007/s10619-008-7030-7.
- Zhemin, D., & Hamdi, M. (2003). Resource management in multi-segment optical networks using the blocking island paradigm. *Workshop on High Performance Switching and Routing, 2003, HPSR*, 43-48. DOI: 10.1109/HPSR.2003.1226678.
- Zhu, Y., Wu, W., & Li, D. (2016). Efficient Client Assignment for Client-Server Systems. *IEEE Transactions on Network and Service Management*. Vol. 13, Issue 4, 835-847. DOI: 10.1109/TNSM.2016.2597269.

Anexo



Descripción de las Pruebas Preliminares

Este anexo presenta los datos utilizados en las Pruebas Preliminares de manera más detallada, así como los resultados obtenidos en cada una de ellas.


1^{ra} Prueba Preliminar: “Segmentación y reconstrucción de recursos contenidos en Servicios Web de Aprendizaje mediante segmentos de distintos tamaños”

Esta prueba consistió en verificar un rango límite para aplicar la actividad de segmentación a los recursos de aprendizaje empaquetados en cada Servicio Web de Aprendizaje, para determinar a partir de que cantidad de tamaño se realizará o no su partición por segmentos, así como el tamaño estándar que tendrá cada uno de estos.

Esta prueba se realizó debido a que los primeros recursos que se intentaron reconstruir con los rangos establecidos para 1,000,000 Bytes (0.95 MB) y 5,000,000 Bytes (4.76 MB), presentaron ciertos daños al momento de reconstruir al recurso en el cliente PHP.

Por lo que se investigó cual era la causa que ocasionaba dicho problema y un nuevo rango que permitiera la correcta reconstrucción de los recursos contenidos en los Servicios Web de Aprendizaje. Los datos utilizados en esta prueba pueden apreciarse en la Tabla A.1.

Tabla A.1. Datos utilizados en la Prueba Preliminar 1

Nombre de la Prueba	Segmentación y reconstrucción de recursos contenidos en Servicios Web de Aprendizaje mediante segmentos de distintos tamaños
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta reconstrucción de los recursos contenidos en Servicios Web de Aprendizaje que han sido segmentados utilizando distintos tamaños en la cantidad de los segmentos.
Precondiciones	<ul style="list-style-type: none"> ▪ Definir un tamaño estándar de los segmentos resultantes. ▪ El componente de software debe funcionar correctamente.
Entrada	<ul style="list-style-type: none"> • Uno de los Recursos de Aprendizaje seleccionados en la Tabla 5.2. • Un rango en el tamaño para los segmentos.
Salida	El recurso contenido por el Servicio Web de Aprendizaje será reconstruido y presentado en la vista del Cliente PHP.
Criterio de Éxito	Cuando el Servicio Web de Aprendizaje es reconstruido de manera correcta, se presentará su recurso contenido en la vista del cliente web:
	 <p>De caso contrario, nos informará que ocurrió un error al momento de la reconstrucción. En caso de que el tamaño de segmentos definidos no sea un número en tercia, se generará código blanco en la cadena Base64 lo que ocasionará que el recurso presente daños o alteraciones al momento de ser presentado.</p>

Se definieron distintos tamaños en la actividad de segmentación y se cambió el tamaño estándar para cada uno de los segmentos resultantes para probar con cuál de ellos se obtenía un menor tiempo de respuesta, tal como se aprecia en la Tabla A.2.

Tabla A.2. Rangos probados para definir el tamaño en la actividad de segmentación

No. de Rango	Tamaño límite para realizar la segmentación		Asignación de tamaño para cada segmento		Generación de Código Blanco	Respuesta de Reconstrucción
	Bytes	Megabytes	Bytes	Megabytes		
1	1,000,000 B	0.95 MB	0 - 500,000 B	0 - 0.47 MB	Si	No exitoso
2	5,000,000 B	4.76 MB	0 - 500,000 B	0 - 0.47 MB	Si	No exitoso
3	9,000,000 B	8.58 MB	0 - 3,000,000 B	0 - 2.86 MB	No	Exitoso
4	10,000,000 B	9.54 MB	0 - 5,000,000 B	0 - 4.76 MB	Si	No exitoso
5	12,000,000 B	11.44 MB	0 - 3,000,000 B	0 - 2.86 MB	No	Exitoso
6	12,000,000 B	11.44 MB	0 - 5,000,000 B	0 - 4.76 MB	Si	No exitoso
7	12,000,000 B	11.44 MB	0 - 6,000,000 B	0 - 5.72 MB	No	Exitoso
8	12,000,000 B	11.44 MB	0 - 9,000,000 B	0 - 8.58 MB	No	Exitoso

Los resultados indicaron que el tamaño en bytes a considerar para realizar la segmentación y sus partes debe estar enfocada en una cantidad de tercias, por ejemplo 3,000,000 Bytes que es igual a 2.86 MB o 6,000,000 Bytes que son 5.72 MB. De lo contrario, se generará código blanco en los segmentos de las cadenas Base64 resultantes y dañará su contenido.

La condición de tener que elegir un número en tercias sucede debido a que debe realizarse una doble conversión del recurso antes de ser segmentado. Esto ocurre debido a que el recurso original tiene entrada principal que se encuentra codificada en una cadena ASCII y son almacenadas en bytes, lo que es idéntico a una cadena binaria en Base2. La cual deberá convertirse ahora a una cadena de tres bytes que conformarán un búfer de 24 bits, el cual será su cadena Base64 resultante [MDN Web Docs, 2013]. Y es por tal motivo, que al indicar en la actividad de segmentación un número en bytes que no sea divisible en tercias (entero divisible entre 3), generará código blanco, lo que ocasionará daños al recurso original al momento de reconstruirlo. Un ejemplo, de la conversión del recurso de entrada a bits y luego la obtención de su cadena en Base64 se puede apreciar en la Tabla A.3.

Tabla A.3. Doble conversión de una cadena ASCII a binario, y de binario a Base64

Texto de entrada ASCII	M								a								n							
	77								97								110							
Bits	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Índice	19				22				5				46											
Resultado en Base64	T				W				F				u											

La Tabla B.3 presenta un ejemplo de la representación mínima de entrada en el esquema de codificación ASCII para ser convertido al esquema de codificación en Base64. Cada carácter en el esquema ASCII representa a un conjunto de 8 caracteres binarios, y cada carácter en el esquema Base64 representa a un conjunto de 6 caracteres binarios. Siendo el valor común

para la conversión de un total de 24 caracteres binarios, 3 caracteres en el esquema ASCII y 4 caracteres en Base64.

Debido a esta doble conversión, que se declara una cantidad en bytes la cual sea un número divisible entre 3, para que la entrada de estos tres bytes sin codificar (en este caso, caracteres de cadena ASCII) den como resultado a cuatro caracteres de la cadena Base64. Obteniendo una correcta segmentación del contenido codificado del recurso original y permita una adecuada reconstrucción sin daños en su cadena de información.

De caso contrario a no definir un tamaño divisible entre 3, se genera “código blanco” el cual busca llenar con “0” al convertir la cadena binaria al esquema Base64. Esto ocasiona que el recurso al ser reconstruido presente daños al ser presentado o no permita visualizarlo.

Para definir a uno de los rangos como la medida estándar para la actividad de segmentación en la estrategia de gestión, se realizaron pruebas en los tiempos de reconstrucción indicados en la Tabla A.4.

Tabla A.4. Resultados en los tiempos de reconstrucción para segmentos de distintos tamaños

No. de Rango	Tamaño del RA	Tiempo de Reconstrucción		Tamaño del RA	Tiempo de Reconstrucción		Tamaño del RA	Tiempo de Reconstrucción	
		SOAP	REST		SOAP	REST		SOAP	REST
1	8.9 MB	1843 ms	2015 ms	12.1 MB	2478 ms	2268 ms	27.9 MB	8647 ms	6984 ms
2	8.9 MB	2088 ms	1964 ms	12.1 MB	2426 ms	2483 ms	27.9 MB	9681 ms	8893 ms
3	8.9 MB	2281 ms	2679 ms	12.1 MB	2616 ms	2827 ms	27.9 MB	10847 ms	9124 ms
4	8.9 MB	2269 ms	2485 ms	12.1 MB	2684 ms	2803 ms	27.9 MB	8657 ms	9246 ms
5	8.9 MB	2676 ms	2618 ms	12.1 MB	3287 ms	3149 ms	27.9 MB	12408 ms	10837 ms
6	8.9 MB	2682 ms	2419 ms	12.1 MB	3108 ms	3497 ms	27.9 MB	11007 ms	9871 ms
7	8.9 MB	2607 ms	2461 ms	12.1 MB	2974 ms	3054 ms	27.9 MB	12374 ms	11187 ms
8	8.9 MB	2643 ms	2387 ms	12.1 MB	2893 ms	3218 ms	27.9 MB	11465 ms	10283 ms

Los rangos 1, 2, 4 y 6 fueron descartados al generar código no deseado ocasionando que el recurso sea dañado al intentar reconstruirse. Y al tener un menor tiempo de respuesta en las pruebas realizadas, se definió al rango 3 con la cantidad de 3,000,000 Bytes (2.86 MB) cuyo tamaño es ideal ya que no generará una gran cantidad de segmentos resultantes y no sobrecargan el límite de almacenamiento dinámico del servidor al importar sus clases.

Por lo que se definió a la actividad de segmentación a partir de los 9,000,000 Bytes (8.58 MB) y en caso de que el recurso sea menor no es necesario aplicar dicha segmentación. Sin embargo, esta medida puede posteriormente cambiarse sin presentar problemas.

2^{da} Prueba Preliminar: “Generación de Servicios Web de Aprendizaje para casos de prueba usando las actividades de la estrategia”

Esta prueba preliminar consiste en la correcta generación de Servicios Web de Aprendizaje para los posteriores casos de prueba, utilizando la estrategia de gestión definida que consiste en la implementación de las actividades de segmentación y reconstrucción. Estos servicios pueden ser generados en SOAP y REST, donde a cada uno se codifica el recurso de aprendizaje en una cadena codificada en Base64 y lo empaqueta como un servicio web. Al producto resultante en este proceso se denomina Servicio Web de Aprendizaje con contenido.

En este trabajo se establece un límite de 8.58 MB, qué al ser igual o mayor a este tamaño, se realiza la segmentación en la cadena Base64 del recurso contenido en el SWA. Se considera como criterio de éxito cuando al generar un Servicio Web de Aprendizaje, aparece una notificación de su generación exitosa. Los datos utilizados para la realización de esta prueba se detallan en la Tabla A.5.

Tabla A.5. Datos utilizados en la Prueba Preliminar 2

Nombre de la Prueba	Generación de Servicios Web de Aprendizaje para casos de prueba usando las actividades de la estrategia de gestión
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta generación de Servicios Web de Aprendizaje a partir de un recurso de aprendizaje y la aplicación de las actividades de gestión definidas para los procesos de gestión y reconstrucción.
Precondiciones	<ul style="list-style-type: none"> ▪ Los campos en el formulario deben ser llenados y válidos. ▪ El componente de software debe funcionar correctamente.
Entrada	Uno de los Recursos de Aprendizaje seleccionados en la Tabla 5.2.
Salida	<ul style="list-style-type: none"> ▪ La generación de archivos Java correspondientes al nuevo Servicio Web de Aprendizaje generado que contiene al recurso empaquetado de manera codificada en Base64. ▪ De ser aplicada la actividad de segmentación, se crearán múltiples clases que contendrán la cadena de información del recurso dividida por segmentos.
Criterio de Éxito	<p>Cuando el Servicio Web de Aprendizaje se genera correctamente aplicando las actividades de gestión definidas, se entregará la siguiente notificación en la vista de la aplicación web:</p> <div style="border: 1px solid #ccc; background-color: #e6ffe6; padding: 5px; text-align: center; margin: 10px auto; width: fit-content;"> LWS created! Close </div> <p>De caso contrario, nos informará que ocurrió un error al momento de la generación.</p>

De los recursos de aprendizaje enlistados en la Tabla 5.2 en el [Capítulo 5](#), se realizaron las pruebas de generación para cada uno ellos donde se cuestionan si se generan de manera correcta considerando sus diferentes formatos, tamaños y tipos de contenido. Así como la correcta aplicación de la actividad de segmentación, donde únicamente a recursos que sean iguales o mayores a los 8.58 MB se le aplicará dicha actividad, pero de ser menores no se aplicará y de ser cualquiera de estos dos posibles resultados el caso se considerará como exitoso. Teniendo como propósito principal, contar con recursos de aprendizaje de distintos

tamaños para analizar su comportamiento al empacarlos como Servicios Web de Aprendizaje. Los resultados obtenidos en esta prueba se detallan en la Tabla A.6.

Tabla A.6. Resultados de generación aplicando la actividad de segmentación

No. del RA	Tamaño	Formato	Tipo	Actividad de Segmentación	Respuesta de la Actividad	Respuesta de Generación
1	0.1 MB	JPG	Objetivo	No aplica	Exitoso	Exitoso
2	0.2 MB	PNG	Objetivo	No aplica	Exitoso	Exitoso
3	0.3 MB	PDF	Objetivo	No aplica	Exitoso	Exitoso
4	0.8 MB	PDF	Actividad	No aplica	Exitoso	Exitoso
5	1.8 MB	PDF	Evaluación	No aplica	Exitoso	Exitoso
6	3.5 MB	PDF	Contenido	No aplica	Exitoso	Exitoso
7	4.2 MB	PPT	Contenido	No aplica	Exitoso	Exitoso
8	5.1 MB	PDF	Contenido	No aplica	Exitoso	Exitoso
9	8.9 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
10	9.0 MB	MP4	Objetivo	Aplica	Exitoso	Exitoso
11	9.1 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
12	9.5 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
13	9.9 MB	MP4	Objetivo	Aplica	Exitoso	Exitoso
14	10.6 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
15	10.8 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
16	11.4 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
17	12.1 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
18	12.6 MB	PPT	Contenido	Aplica	Exitoso	Exitoso
19	13.8 MB	DOC	Contenido	Aplica	Exitoso	Exitoso
20	14.6 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
21	16.8 MB	PPT	Contenido	Aplica	Exitoso	Exitoso
22	17.1 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
23	18.3 MB	DOC	Contenido	Aplica	Exitoso	Exitoso
24	25.2 MB	PPT	Contenido	Aplica	Exitoso	Exitoso
25	27.4 MB	DOC	Contenido	Aplica	Exitoso	Exitoso
26	27.9 MB	MP3	Contenido	Aplica	Exitoso	Exitoso
27	32.5 MB	MP4	Contenido	Aplica	Exitoso	Exitoso
28	42.7 MB	DOC	Contenido	Aplica	Exitoso	Exitoso
29	50.4 MB	PPT	Contenido	Aplica	Exitoso	Exitoso
30	61.1 MB	DOC	Contenido	Aplica	Exitoso	Exitoso
31	67.2 MB	PPT	Contenido	Aplica	Exitoso	Exitoso
32	78.2 MB	MP4	Actividad	Aplica	Exitoso	Exitoso
33	95.6 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
34	98.8 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
35	133.5 MB	MP4	Contenido	Aplica	Exitoso	Exitoso
36	293.4 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
37	314.0 MB	PDF	Contenido	Aplica	Exitoso	Exitoso
38	409.9 MB	MP4	Contenido	Aplica	Exitoso	Exitoso
39	528.3 MB	MP4	Contenido	Aplica	Exitoso	Exitoso

De acuerdo a los resultados obtenidos en la Tabla A.6 se muestra que se cumple con la actividad de segmentación, cuya condición únicamente se aplica cuando el tamaño del recurso de aprendizaje es igual o mayor a los 8.58 MB. Mientras que, para recursos de menor tamaño a este límite, se genera el Servicio Web de Aprendizaje empaquetando al recurso sin segmentarlo.

Anexo

B

Descripción de las Interfaces

Este anexo presenta las interfaces desarrolladas en este trabajo de la Herramienta Java y del Cliente PHP de manera más detallada, así como la funcionalidad en cada una de sus vistas.

Interfaces de la herramienta Java: “Generación y Composición”

La herramienta Java se encarga de la generación de Servicios Web de Aprendizaje mediante la segmentación de los recursos de aprendizaje empaquetados en estos. De igual manera, la herramienta permite realizar la composición de múltiples SWA para poder generar a un Servicio Web de Aprendizaje Compuesto, el cual contenga un arreglo de invocaciones de al menos dos SWA que han sido segmentados.

La Figura B.1 corresponde a la vista inicial de la herramienta Java, cuyo enlace es “https://localhost:8084/LWS_v2/index.jsp” donde se presentan las siguientes secciones:

1. Es el enlace que redirecciona a la vista inicial de la herramienta, denominada “LWS” por ser las siglas en inglés de Servicios Web de Aprendizaje (Learning Web Services).
2. El menú desplegable “JAEM”, corresponde al módulo de la herramienta antecedente “LWS” [Escobar Megchún, 2017].
3. El menú desplegable “IHFC”, corresponde al módulo que contiene a la extensión implementada en este trabajo.
4. Es el título de la herramienta “Web Application to create and compose Learning Web Services”.
5. Es el título en inglés del trabajo de tesis antecedente “Generator of Learning Web Services Composed from Educational Resources” [Escobar Megchún, 2017].
6. Es el título en inglés de este trabajo de tesis “Management of Learning Resources to generate Learning Web Services”.
7. Es el pie de página que indica la pertenencia intelectual a nombre del CENIDET y sus autores.

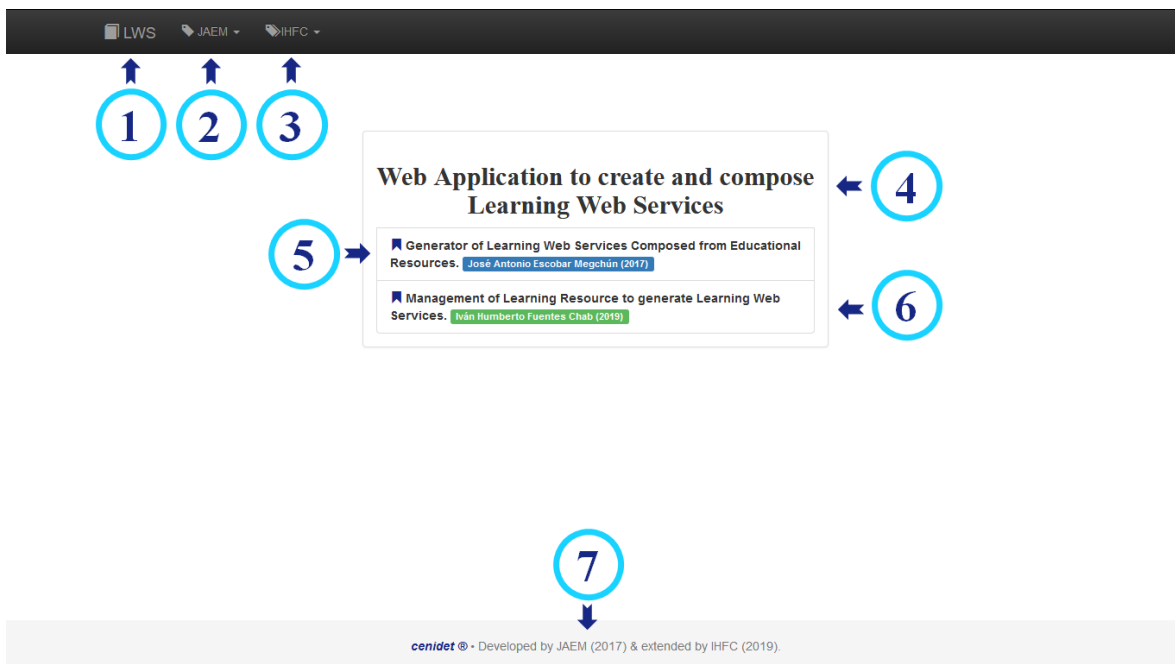


Figura B.1. Vista inicial de la herramienta Java

La Figura B.2 corresponde a la vista encargada de la generación de Servicios Web de Aprendizaje a partir de un recurso de aprendizaje, donde se aplica la estrategia de gestión para empaquetar recursos codificados en servicios web y segmentarlos en caso de ser necesario, cuyo enlace es “https://localhost:8084/LWS_v2/createSegmented.jsp” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.
3. Es un campo de texto para asignarle el nombre al SWA a generar.
4. Se indica si el recurso a ingresar es desde un archivo o referenciado externamente a través de una URL.
5. Es el botón para cargar al recurso que contendrá el servicio.
6. Es el tipo de servicio a generar, pudiendo ser SOAP o REST.
7. El botón “Generate LWS by Segments” es la acción de codificar al recurso ingresado, y aplicar la estrategia de gestión donde únicamente se aplicará la actividad de segmentación y las condiciones para su reconstrucción en caso de ser necesario, para entregar los archivos Java correspondientes al Servicio Web de Aprendizaje creado.

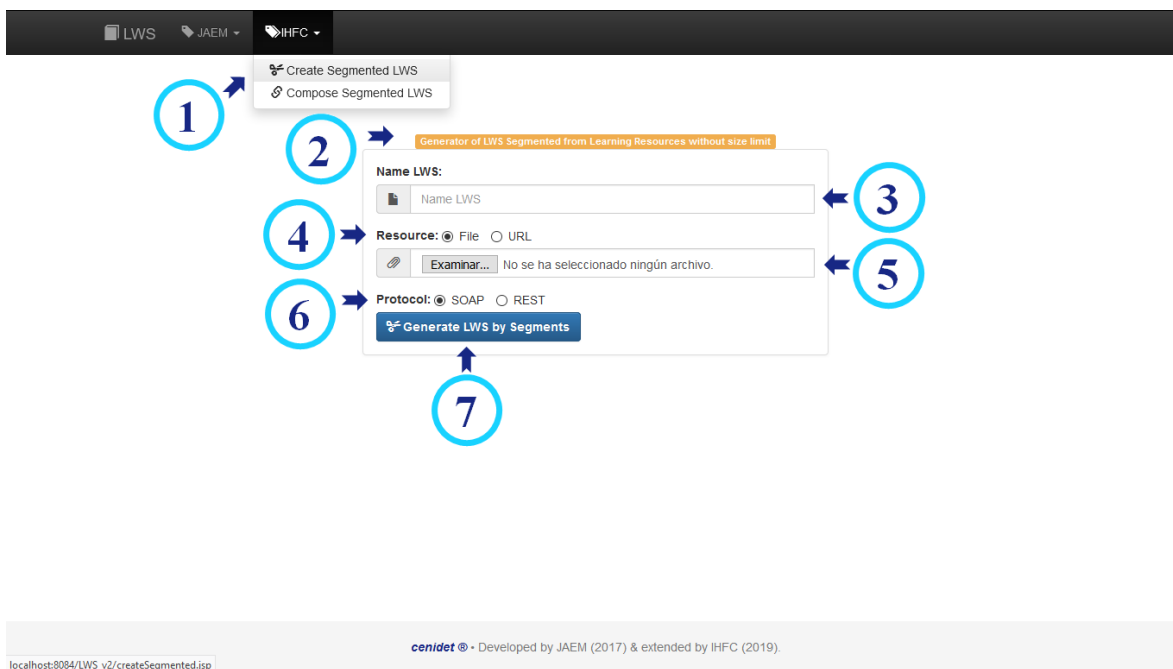


Figura B.2. Vista para la generación de un SWA aplicando la estrategia de gestión

La Figura B.3 corresponde a la vista encargada de realizar la composición de al menos dos Servicios Web de Aprendizaje cuyos recursos empaquetados han sido segmentados, cuyo enlace es “https://localhost:8084/LWS_v2/composeSegmented.jsp” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.

3. El botón “Add Url” permite añadir otro campo de texto como el indicado en el punto (6) para ingresar otro SWA.
4. Es un campo de texto para asignarle el nombre al SWA-C a generar.
5. Es un campo para ingresar el primer SWA, el cual, aunque puede ser de cualquier tipo, se recomienda que sea de tipo Objetivo porque es el que dará forma al contenido educativo que se presentará al usuario.
6. Es un campo para ingresar otro SWA, pudiendo ser alguno de los tipos: Contenido, Actividad y/o Evaluación.
7. El botón “Compose LWS by Segments” es la acción de crear un archivo Java, el cual contenga un arreglo de invocaciones con todos los SWA en el orden indicado por el usuario.

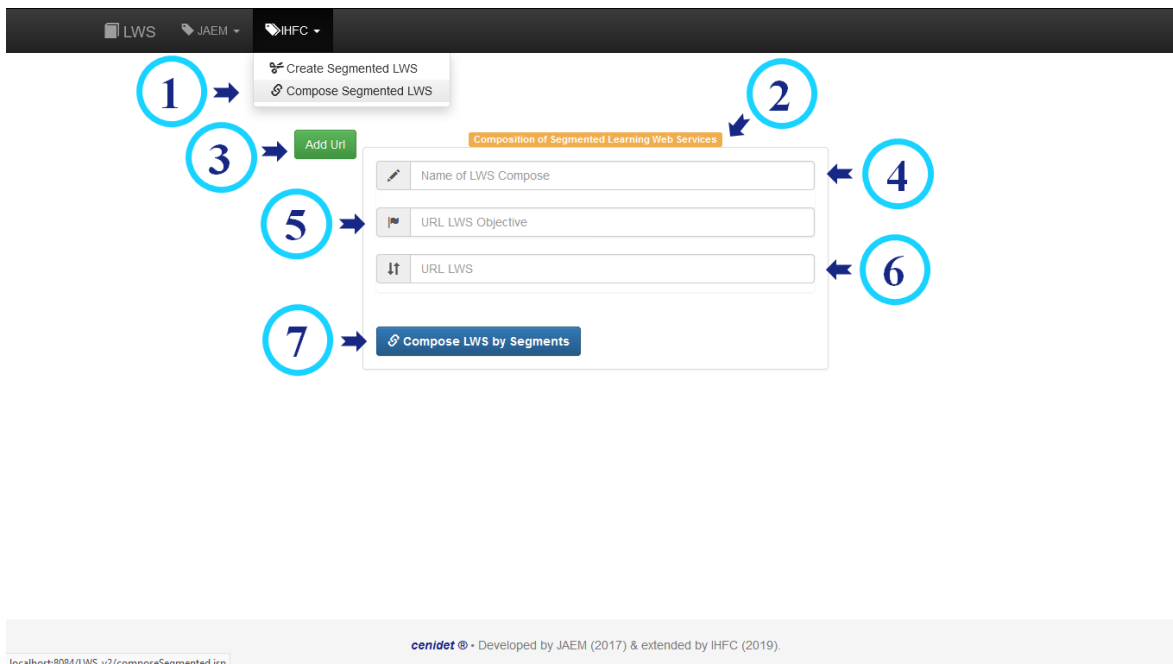


Figura B.3. Vista para la composición de SWA con contenidos segmentados

Interfaces del cliente PHP: “Reconstrucción y Presentación”

El cliente PHP se encarga de la reconstrucción de los recursos empaquetados que han sido segmentados en un Servicio Web de Aprendizaje, para poder presentarlos de manera correcta. De igual manera, el cliente permite presentar los recursos de SWA tanto atómicos como compuestos en SOAP y REST.

La Figura B.4 corresponde a la vista inicial del cliente PHP, cuyo enlace es “https://localhost/LWS-client_v2/index.php” donde se presentan las siguientes secciones:

1. Es el enlace que redirecciona a la vista inicial del cliente, denominada “LWS Client” por ser las siglas en inglés de Cliente de Servicios Web de Aprendizaje (Learning Web Services Client).

2. El menú despegable “JAEM”, corresponde al módulo del cliente antecedente “clientSoap” [Escobar Megchún, 2017].
3. El menú despegable “IHFC”, corresponde al módulo que contiene a la extensión implementada en este trabajo.
4. Es el título del cliente “Web Client to rebuild and present Learning Web Services”.
5. Es el título en inglés del trabajo de tesis antecedente “Generator of Learning Web Services Composed from Educational Resources” [Escobar Megchún, 2017].
6. Es el título en inglés de este trabajo de tesis “Management of Learning Resources to generate Learning Web Services”.
7. Es el pie de página que indica la pertenencia intelectual a nombre del CENIDET y sus autores.

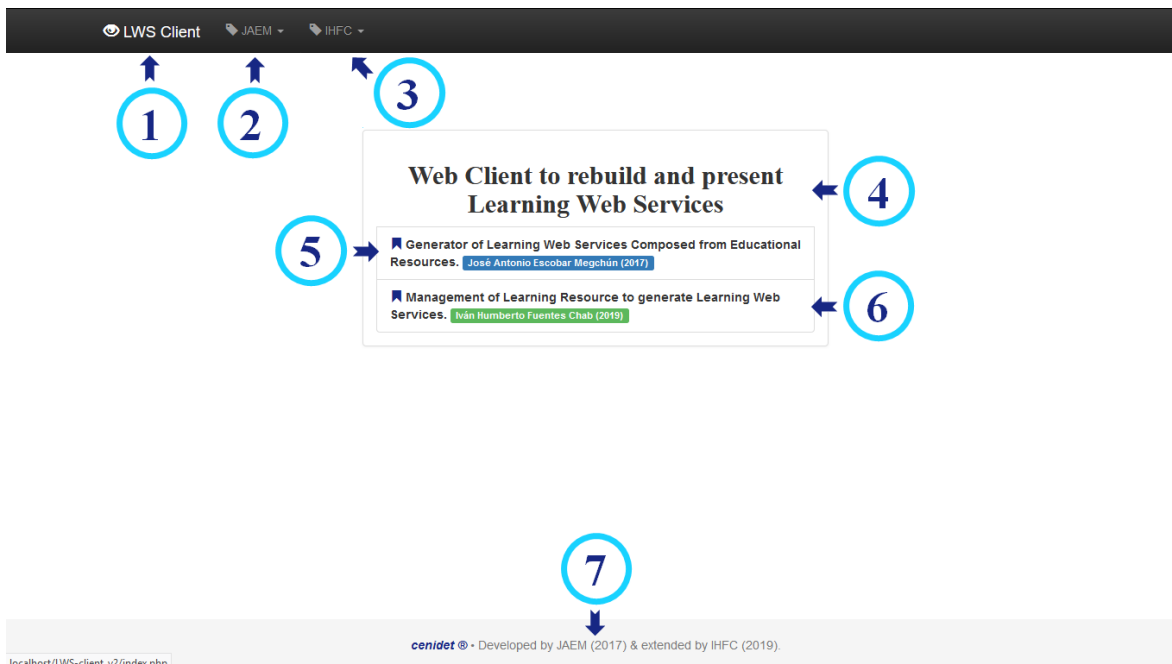


Figura B.4. Vista inicial del cliente PHP

La Figura B.5 corresponde a la vista encargada de reconstruir y presentar un recurso segmentado empaquetado en un SWA de tipo SOAP, cuyo enlace es “https://localhost/LWS-client_v2/testBig.php” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.
3. Es un campo de texto para ingresar el SWA de tipo SOAP a presentar.
4. El botón “Test SOAP Segmented LWS” es la acción de presentar en la vista del navegador web el recurso de aprendizaje contenido en el SWA, y de reconstruirlo en caso de encontrarse segmentado para poder ser visualizado.
5. Es un aviso que indica al usuario que debe ingresar el SWA de tipo SOAP.

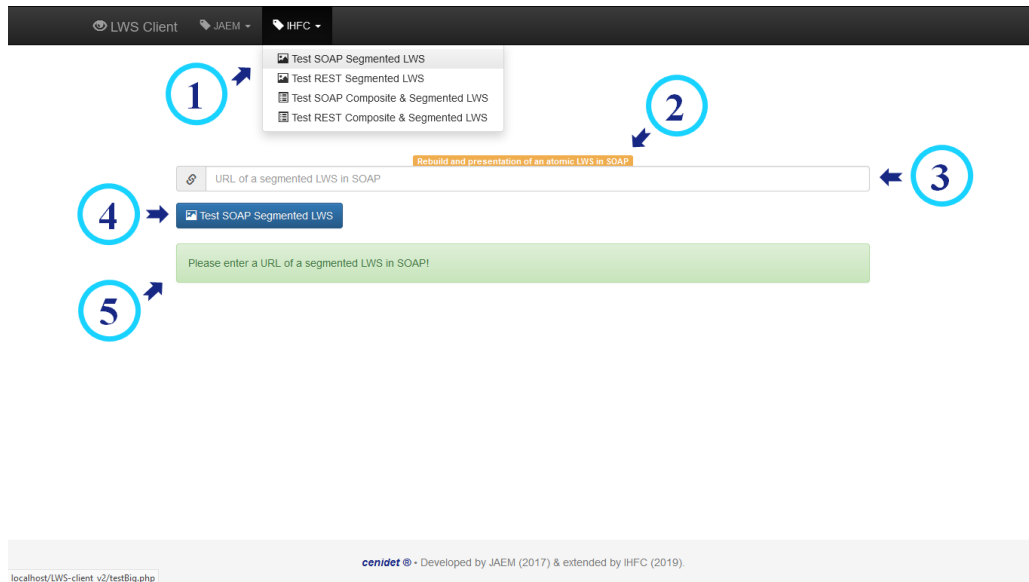


Figura B.5. Vista para la presentación de un SWA con contenido segmentado en SOAP

La Figura B.6 corresponde a la vista encargada de reconstruir y presentar un recurso segmentado empaquetado en un SWA de tipo REST, cuyo enlace es “https://localhost/LWS-client_v2/testRestBig.php” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.
3. Es un campo de texto para ingresar el SWA de tipo REST a presentar.
4. El botón “Test REST Segmented LWS” es la acción de presentar en la vista del navegador web el recurso de aprendizaje contenido en el SWA, y de reconstruirlo en caso de encontrarse segmentado para poder ser visualizado.
5. Es un aviso que indica al usuario que debe ingresar el SWA de tipo REST.

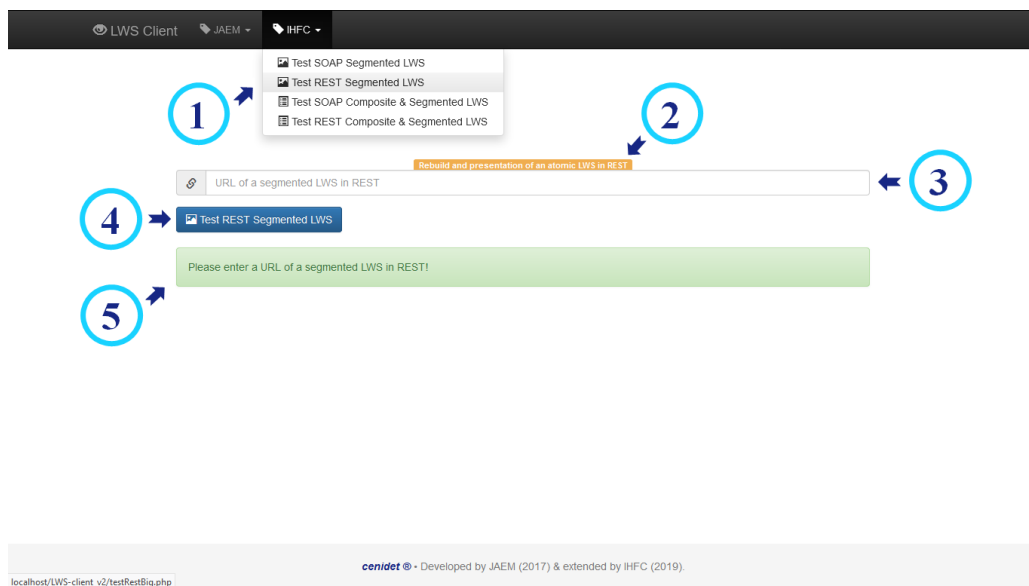


Figura B.6. Vista para la presentación de un SWA con contenido segmentado en REST

La Figura B.7 corresponde a la vista encargada de reconstruir y presentar múltiples recursos que han sido segmentados y se encuentran empaquetados en un Servicio Web de Aprendizaje Compuesto con el protocolo SOAP, cuyo enlace es “https://localhost/LWS-client_v2/testCompBig.php” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.
3. Es un campo de texto para ingresar el SWA-C de tipo SOAP a presentar.
4. El botón “Test SOAP Composite & Segmented LWS” es la acción de presentar en la vista del navegador web todos los recursos de aprendizaje contenidos en el SWA-C, y de reconstruir a cada recurso que se encuentre segmentado para ser visualizado.
5. Es un aviso que indica al usuario que debe ingresar el SWA-C de tipo SOAP.

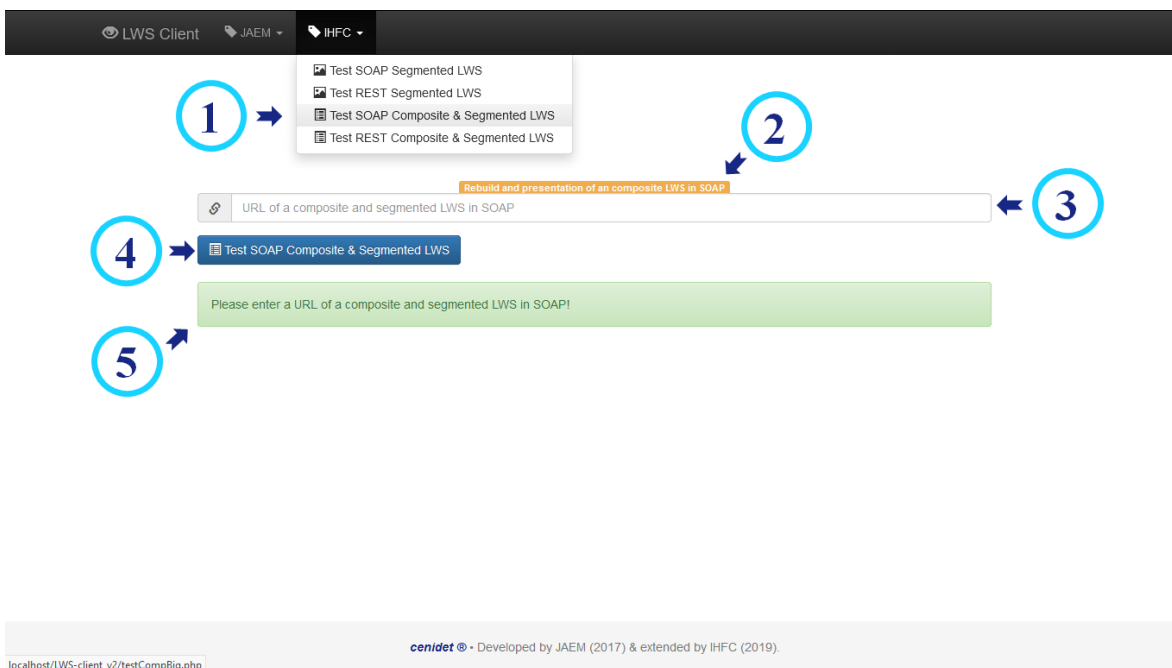


Figura B.7. Vista para la presentación de un SWA-C con contenidos segmentados en SOAP

La Figura B.8 corresponde a la vista encargada de reconstruir y presentar múltiples recursos que han sido segmentados y se encuentran empaquetados en un Servicio Web de Aprendizaje Compuesto con la arquitectura REST, cuyo enlace es “https://localhost/LWS-client_v2/testCompRestBig.php” donde se presentan las siguientes secciones:

1. Es el enlace que corresponde a esta vista.
2. Es una leyenda que indica el funcionamiento de la vista.
3. Es un campo de texto para ingresar el SWA-C de tipo REST a presentar.
4. El botón “Test REST Composite & Segmented LWS” es la acción de presentar en la vista del navegador web todos los recursos de aprendizaje contenidos en el SWA-C, y de reconstruir a cada recurso que se encuentre segmentado para ser visualizado.
5. Es un aviso que indica al usuario que debe ingresar el SWA-C de tipo REST.

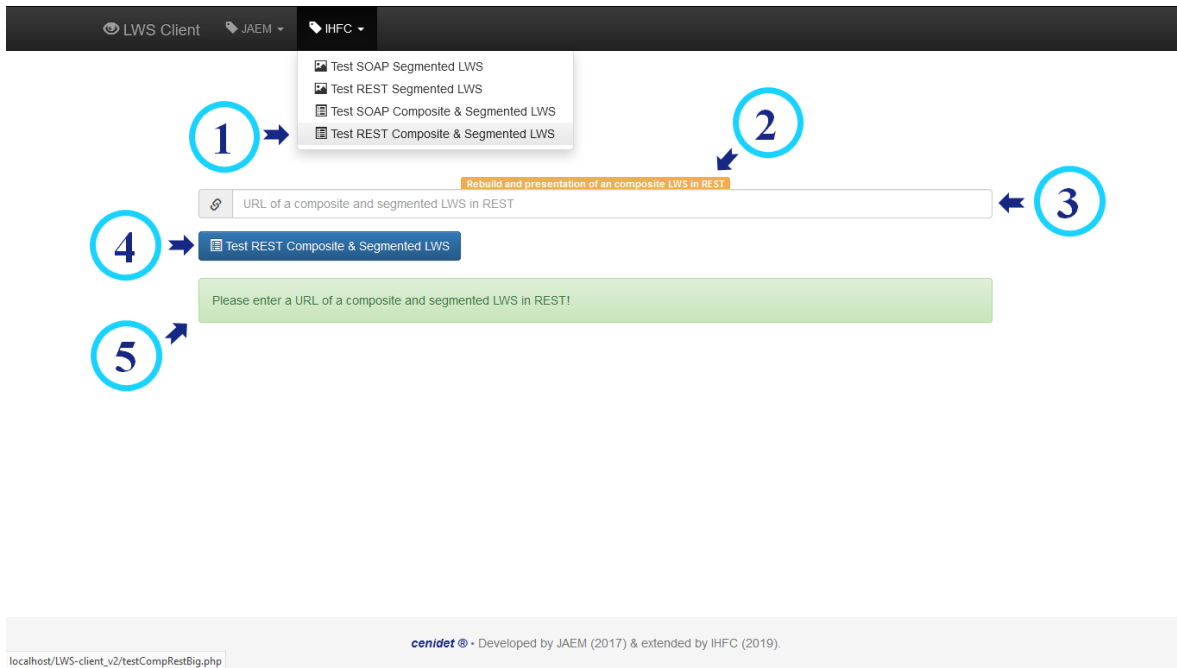


Figura B.8. Vista para la presentación de un SWA-C con contenidos segmentados en REST

Anexo

C

Descripción de los Tipos de Pruebas

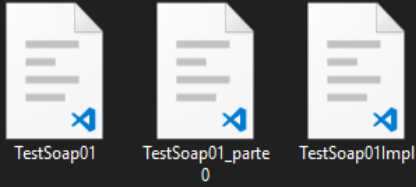
Este anexo presenta los datos utilizados en los Tipos de Pruebas de manera más detallada, así como los resultados obtenidos en cada uno de ellos.

1^{er} Tipo de Prueba: “Segmentación de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje”

De los 39 recursos de aprendizaje enlistados en la Tabla 5.2 en el Capítulo 5, se realiza la generación de Servicios Web de Aprendizaje con SOAP y REST donde se realizan pruebas en la segmentación, para identificar la cantidad de segmentos resultantes, la diferencia entre el tamaño original del recurso y del SWA con contenido, y mediante el entorno de pruebas de Apache JMeter se obtiene el tiempo de respuesta. Los datos utilizados para la realización de esta prueba se detallan en la Tabla C.1.

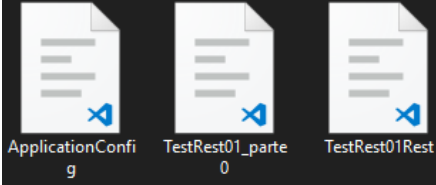
Tabla C.1. Datos utilizados en el Tipo de Prueba 1

Nombre de la Prueba	Segmentación de Recursos de Aprendizaje para generar Servicios Web de Aprendizaje
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta segmentación y ordenamiento de las cadenas en Base64 de los recursos empaquetados como Servicios Web de Aprendizaje a través de SOAP y REST.
Precondiciones	<ul style="list-style-type: none"> ▪ Los campos en el formulario deben ser llenados y válidos. ▪ El componente de software debe funcionar correctamente.
Entrada	Uno de los Recursos de Aprendizaje seleccionados de la Tabla 5.2.
Salida	<ul style="list-style-type: none"> ▪ La generación de archivos Java correspondientes al nuevo Servicio Web de Aprendizaje contiene al recurso empaquetado de manera codificada en Base64. ▪ De ser aplicada la actividad de segmentación, se crearán múltiples clases que contendrán la cadena de información del recurso dividida por segmentos. ▪ El tiempo de respuesta para realizar la segmentación del recurso contenido en el Servicio Web de Aprendizaje. ▪ Para el protocolo SOAP: <ul style="list-style-type: none"> ○ Se generan al menos tres archivos Java: <ul style="list-style-type: none"> ▪ “NombreSWA.java” correspondiente a la interfaz del Servicio Web de Aprendizaje. ▪ “NombreSWAImpl.java” correspondiente a los métodos de implementación del Servicio Web de Aprendizaje. ▪ “NombreSWA_parte0.java” correspondiente al contenido del recurso codificado en base64. ▪ Para el protocolo REST: <ul style="list-style-type: none"> ○ Se generan al menos tres archivos Java: <ul style="list-style-type: none"> ▪ “NombreSWARest.java” correspondiente al Servicio Web de Aprendizaje. ▪ “ApplicationConfig.java” correspondiente al archivo de configuración para el despliegue del Servicio Web de Aprendizaje. ▪ “NombreSWA_parte0.java” correspondiente al contenido del recurso codificado en base64. ▪ Este último archivo, tanto para SOAP como para REST, puede llegar hasta el número de segmento n siendo nombrado “NombreSWA_parten.java”, siendo n el número final del total de segmentos obtenidos. Creando un archivo parte para cada segmento en que se haya dividido el recurso contenido.
Criterio de Éxito	<ul style="list-style-type: none"> ▪ Cuando el Servicio Web de Aprendizaje se genera correctamente mediante segmentación en el caso de SOAP: <ul style="list-style-type: none"> ○ Se generan los archivos Java:



- En este caso el recurso contenido no es mayor a los 8.58 MB, por lo que solo se genera “TestSoap01_parte0.java”.
- En caso contrario que sea igual o mayor a los 8.58 MB, se generarán tantos archivos partes hasta llegar a “TestSoap01_parten.java”, siendo *n* el número final del total de segmentos obtenidos.

- Cuando el Servicio Web de Aprendizaje se genera correctamente mediante segmentación en el caso de REST:
 - Se generan los archivos Java:


 - En este caso el recurso contenido no es mayor a los 8.58 MB, por lo que solo se genera “TestRest01_parte0.java”.
 - En caso contrario que sea igual o mayor a los 8.58 MB, se generarán tantos archivos partes hasta llegar a “TestRest01_parten.java”, siendo *n* el número final del total de segmentos obtenidos.

La cantidad de segmentos obtenidos puede obtenerse con los archivos Java generados para cada Servicio Web de Aprendizaje, mientras que los tiempos de respuesta son obtenidos de acuerdo al entorno de pruebas al momento de su generación. Los resultados obtenidos en esta prueba se detallan en la Tabla C.2. Donde los recursos del 1 al 8 presentan únicamente un segmento, debido a que no fueron segmentados.

Tabla C.2. Resultados de segmentación de RA para generar SWA en SOAP y REST

No. del RA	Tamaño del RA	Formato	Tipo	Protocolo	Tamaño del SWA Generado	Cantidad de Segmentos	Tiempo de Segmentación
1	0.1 MB	JPG	Objetivo	SOAP	0.23 MB	1	268 ms
				REST	0.23 MB	1	243 ms
2	0.2 MB	PNG	Objetivo	SOAP	0.32 MB	1	296 ms
				REST	0.32 MB	1	229 ms
3	0.3 MB	PDF	Objetivo	SOAP	0.45 MB	1	298 ms
				REST	0.45 MB	1	274 ms
4	0.8 MB	PDF	Actividad	SOAP	1.05 MB	1	764 ms
				REST	1.05 MB	1	726 ms
5	1.8 MB	PDF	Evaluación	SOAP	2.48 MB	1	702 ms
				REST	2.48 MB	1	644 ms
6	3.5 MB	PDF	Contenido	SOAP	4.78 MB	1	1209 ms
				REST	4.78 MB	1	1018 ms
7	4.2 MB	PPT	Contenido	SOAP	5.62 MB	1	1862 ms
				REST	5.62 MB	1	1803 ms
8	5.1 MB	PDF	Contenido	SOAP	6.77 MB	1	2104 ms
				REST	6.77 MB	1	2037 ms

9	8.9 MB	PDF	Contenido	SOAP	11.6 MB	4	2026 ms
				REST	11.6 MB	4	1981 ms
10	9.0 MB	MP4	Objetivo	SOAP	11.8 MB	4	1802 ms
				REST	11.8 MB	4	1798 ms
11	9.1 MB	PDF	Contenido	SOAP	11.9 MB	4	2643 ms
				REST	11.9 MB	4	1998 ms
12	9.5 MB	PDF	Contenido	SOAP	12.3 MB	4	2682 ms
				REST	12.3 MB	4	2479 ms
13	9.9 MB	MP4	Objetivo	SOAP	12.9 MB	4	2624 ms
				REST	12.9 MB	4	2616 ms
14	10.6 MB	PDF	Contenido	SOAP	13.9 MB	4	2078 ms
				REST	13.9 MB	4	1874 ms
15	10.8 MB	PDF	Contenido	SOAP	14.0 MB	4	2942 ms
				REST	14.0 MB	4	2891 ms
16	11.4 MB	PDF	Contenido	SOAP	14.8 MB	4	2865 ms
				REST	14.8 MB	4	2632 ms
17	12.1 MB	PDF	Contenido	SOAP	15.7 MB	5	2897 ms
				REST	15.7 MB	5	2644 ms
18	12.6 MB	PPT	Contenido	SOAP	16.4 MB	5	3024 ms
				REST	16.4 MB	5	2876 ms
19	13.8 MB	DOC	Contenido	SOAP	18.0 MB	5	3209 ms
				REST	18.0 MB	5	3043 ms
20	14.6 MB	PDF	Contenido	SOAP	18.9 MB	5	3245 ms
				REST	18.9 MB	5	3189 ms
21	16.8 MB	PPT	Contenido	SOAP	21.8 MB	6	3086 ms
				REST	21.8 MB	6	3002 ms
22	17.1 MB	PDF	Contenido	SOAP	22.1 MB	6	3207 ms
				REST	22.1 MB	6	2976 ms
23	18.3 MB	DOC	Contenido	SOAP	23.8 MB	6	3427 ms
				REST	23.8 MB	6	3219 ms
24	25.2 MB	PPT	Contenido	SOAP	32.8 MB	10	6482 ms
				REST	32.8 MB	10	6297 ms
25	27.4 MB	DOC	Contenido	SOAP	35.6 MB	10	7415 ms
				REST	35.6 MB	10	6473 ms
26	27.9 MB	MP3	Contenido	SOAP	36.3 MB	10	6218 ms
				REST	36.3 MB	10	5893 ms
27	32.5 MB	MP4	Contenido	SOAP	42.4 MB	12	6874 ms
				REST	42.4 MB	12	5237 ms
28	42.7 MB	DOC	Contenido	SOAP	55.5 MB	16	9492 ms
				REST	55.5 MB	16	8713 ms
29	50.4 MB	PPT	Contenido	SOAP	65.5 MB	18	11785 ms
				REST	65.5 MB	18	10682 ms
30	61.1 MB	DOC	Contenido	SOAP	79.3 MB	24	14307 ms
				REST	79.3 MB	24	14189 ms
31	67.2 MB	PPT	Contenido	SOAP	87.4 MB	25	19805 ms
				REST	87.4 MB	25	18997 ms
32	78.2 MB	MP4	Actividad	SOAP	101.0 MB	27	20709 ms
				REST	101.0 MB	27	19891 ms
33	95.6 MB	PDF	Contenido	SOAP	124.0 MB	33	24386 ms
				REST	124.0 MB	33	24174 ms
34	98.8 MB	PDF	Contenido	SOAP	128.0 MB	34	24963 ms
				REST	128.0 MB	34	24615 ms
35	133.5 MB	MP4	Contenido	SOAP	173.0 MB	46	40187 ms
				REST	173.0 MB	46	40071 ms

36	293.4 MB	PDF	Contenido	SOAP	382.0 MB	101	82349 ms
				REST	382.0 MB	101	81077 ms
37	314.0 MB	PDF	Contenido	SOAP	409.0 MB	108	91147 ms
				REST	409.0 MB	108	86979 ms
38	409.9 MB	MP4	Contenido	SOAP	546.0 MB	144	163789 ms
				REST	546.0 MB	144	156087 ms
39	528.3 MB	MP4	Contenido	SOAP	688.0 MB	181	218905 ms
				REST	688.0 MB	181	206348 ms

Las pruebas realizadas en cada recurso de aprendizaje, indican que los cortes a sus cadenas Base64 asignan segmentos de tamaño uniforme de manera correcta, los tiempos de respuesta para este proceso son distintos dependiendo del tamaño del recurso contenido en cada Servicio Web de Aprendizaje, el cual fue obtenido del promedio de la ejecución de 10 usuarios en un lapso de 5 segundos con el entorno de pruebas Apache JMeter.

Los resultados fueron los mismos en el tamaño generado del SWA con contenido tanto para SOAP como para REST, el cual aumenta un aproximado del 30% del tamaño original del recurso, esto ocasionado al codificarlo en grandes cadenas de texto de Base64 y empaquetarlo en archivos Java. La cantidad de segmentos resultantes fue la misma en ambos protocolos, lo que indica que la actividad de segmentación y los límites establecidos funcionan correctamente. Los resultados variaron únicamente en los tiempos de segmentación, siendo con la arquitectura REST donde se obtuvieron el menor tiempo de respuesta, aunque cabe mencionar que los tiempos pueden variar dependiendo de las configuraciones del equipo y del entorno de pruebas.

2^{do} Tipo de Prueba: “Presentación de Servicios Web de Aprendizaje con contenido segmentado”

De los 78 SWA generados en el Primer Tipo de Prueba y que se enlistan en la Tabla C.2, se realizan las pruebas de reconstrucción con SOAP y REST, para analizar si sus tamaños corresponden al tamaño original del recurso al ser reconstruido, así como su correcta presentación de manera visual, y mediante el entorno de pruebas de Apache JMeter se obtiene un tiempo promedio de respuesta. Los datos utilizados para la realización de esta prueba se detallan en la Tabla C.3.

Tabla C.3. Datos utilizados en el Tipo de Prueba 2

Nombre de la Prueba	Presentación de Servicios Web de Aprendizaje con contenido segmentado
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta reconstrucción de Servicios Web de Aprendizaje que han sido segmentados para que tengan una adecuada presentación a través de SOAP y REST.
Precondiciones	<ul style="list-style-type: none"> ▪ El campo en el formulario debe ser llenado y válido. ▪ El componente de software debe funcionar correctamente. ▪ El servicio web de aprendizaje atómico debe estar alojado en servidor.
Entrada	Uno de los Servicios Web de Aprendizaje generados en el Tipo de Prueba 1.

Salida

El recurso contenido por el Servicio Web de Aprendizaje será reconstruido y presentado en la vista del Cliente PHP.

Cuando el Servicio Web de Aprendizaje es reconstruido de manera correcta, se presentará su recurso de aprendizaje contenido en la vista del cliente web:



Criterio de Éxito

De caso contrario, nos informará que ocurrió un error al momento de la reconstrucción.

Para que los recursos puedan ser presentados de manera adecuada, estos deben ser reconstruidos en el orden correcto sin presentar daños. Para ello, se tomaron los Servicios Web de Aprendizaje generados en el Tipo de Prueba 1, donde se compara el tamaño original del recurso y su tamaño resultante luego de ser segmentado, reconstruido y descargado, así como su tiempo de reconstrucción obtenido con el entorno de pruebas Apache JMeter y las respuestas esperadas al ser reconstruidas y presentadas por el cliente PHP. Los resultados obtenidos en esta prueba mediante SOAP y REST son indicados en la Tabla C.4.

Tabla C.4. Resultados de reconstrucción y presentación con SOAP y REST

No. del SWA	Tamaño del RA Original	Protocolo	Tamaño del SWA Generado	Tamaño del RA Reconstruido	Tiempo de Reconstrucción y Presentación	Respuesta de Reconstrucción	Respuesta de Presentación
1	0.1 MB	SOAP	0.23 MB	0.1 MB	286 ms	No aplica	Exitoso
		REST	0.23 MB	0.1 MB	204 ms	No aplica	Exitoso
2	0.2 MB	SOAP	0.32 MB	0.2 MB	294 ms	No aplica	Exitoso
		REST	0.32 MB	0.2 MB	238 ms	No aplica	Exitoso
3	0.3 MB	SOAP	0.45 MB	0.3 MB	418 ms	No aplica	Exitoso
		REST	0.45 MB	0.3 MB	394 ms	No aplica	Exitoso
4	0.8 MB	SOAP	1.05 MB	0.8 MB	682 ms	No aplica	Exitoso
		REST	1.05 MB	0.8 MB	641 ms	No aplica	Exitoso
5	1.8 MB	SOAP	2.48 MB	1.8 MB	1242 ms	No aplica	Exitoso
		REST	2.48 MB	1.8 MB	1198 ms	No aplica	Exitoso
6	3.5 MB	SOAP	4.66 MB	3.5 MB	1688 ms	No aplica	Exitoso
		REST	4.66 MB	3.5 MB	1609 ms	No aplica	Exitoso
7	4.2 MB	SOAP	5.49 MB	4.2 MB	1672 ms	No aplica	Exitoso
		REST	5.49 MB	4.2 MB	1436 ms	No aplica	Exitoso
8	5.1 MB	SOAP	6.60 MB	5.1 MB	1587 ms	No aplica	Exitoso
		REST	6.60 MB	5.1 MB	1422 ms	No aplica	Exitoso
9	8.9 MB	SOAP	11.6 MB	8.9 MB	2010 ms	Exitoso	Exitoso
		REST	11.6 MB	8.9 MB	1934 ms	Exitoso	Exitoso
10	9.0 MB	SOAP	11.8 MB	9.0 MB	2806 ms	Exitoso	Exitoso
		REST	11.8 MB	9.0 MB	2624 ms	Exitoso	Exitoso
11	9.1 MB	SOAP	11.9 MB	9.1 MB	2418 ms	Exitoso	Exitoso
		REST	11.9 MB	9.1 MB	2053 ms	Exitoso	Exitoso
12	9.5 MB	SOAP	12.3 MB	9.5 MB	2601 ms	Exitoso	Exitoso
		REST	12.3 MB	9.5 MB	2343 ms	Exitoso	Exitoso
13	9.9 MB	SOAP	12.9 MB	9.9 MB	2488 ms	Exitoso	Exitoso
		REST	12.9 MB	9.9 MB	2462 ms	Exitoso	Exitoso

14	10.6 MB	SOAP	13.9 MB	10.6 MB	2807 ms	Exitoso	Exitoso
		REST	13.9 MB	10.6 MB	2636 ms	Exitoso	Exitoso
15	10.8 MB	SOAP	14.0 MB	10.8 MB	2845 ms	Exitoso	Exitoso
		REST	14.0 MB	10.8 MB	2602 ms	Exitoso	Exitoso
16	11.4 MB	SOAP	14.8 MB	11.4 MB	2650 ms	Exitoso	Exitoso
		REST	14.8 MB	11.4 MB	2513 ms	Exitoso	Exitoso
17	12.1 MB	SOAP	15.7 MB	12.1 MB	2674 ms	Exitoso	Exitoso
		REST	15.7 MB	12.1 MB	2566 ms	Exitoso	Exitoso
18	12.6 MB	SOAP	16.4 MB	12.6 MB	2987 ms	Exitoso	Exitoso
		REST	16.4 MB	12.6 MB	2845 ms	Exitoso	Exitoso
19	13.8 MB	SOAP	18.0 MB	13.8 MB	2892 ms	Exitoso	Exitoso
		REST	18.0 MB	13.8 MB	2687 ms	Exitoso	Exitoso
20	14.6 MB	SOAP	18.9 MB	14.6 MB	4861 ms	Exitoso	Exitoso
		REST	18.9 MB	14.6 MB	4212 ms	Exitoso	Exitoso
21	16.8 MB	SOAP	21.8 MB	16.8 MB	4866 ms	Exitoso	Exitoso
		REST	21.8 MB	16.8 MB	4611 ms	Exitoso	Exitoso
22	17.1 MB	SOAP	22.1 MB	17.1 MB	5126 ms	Exitoso	Exitoso
		REST	22.1 MB	17.1 MB	4871 ms	Exitoso	Exitoso
23	18.3 MB	SOAP	23.8 MB	18.3 MB	5653 ms	Exitoso	Exitoso
		REST	23.8 MB	18.3 MB	5598 ms	Exitoso	Exitoso
24	25.2 MB	SOAP	32.8 MB	25.2 MB	8431 ms	Exitoso	Exitoso
		REST	32.8 MB	25.2 MB	7906 ms	Exitoso	Exitoso
25	27.4 MB	SOAP	35.6 MB	27.4 MB	8589 ms	Exitoso	Exitoso
		REST	35.6 MB	27.4 MB	8052 ms	Exitoso	Exitoso
26	27.9 MB	SOAP	36.3 MB	27.9 MB	10847 ms	Exitoso	Exitoso
		REST	36.3 MB	27.9 MB	9124 ms	Exitoso	Exitoso
27	32.5 MB	SOAP	42.4 MB	32.5 MB	12037 ms	Exitoso	Exitoso
		REST	42.4 MB	32.5 MB	10987 ms	Exitoso	Exitoso
28	42.7 MB	SOAP	55.5 MB	42.7 MB	12047 ms	Exitoso	Exitoso
		REST	55.5 MB	42.7 MB	11789 ms	Exitoso	Exitoso
29	50.4 MB	SOAP	65.5 MB	50.4 MB	14815 ms	Exitoso	Exitoso
		REST	65.5 MB	50.4 MB	14667 ms	Exitoso	Exitoso
30	61.1 MB	SOAP	79.3 MB	61.1 MB	23327 ms	Exitoso	Exitoso
		REST	79.3 MB	61.1 MB	21341 ms	Exitoso	Exitoso
31	67.2 MB	SOAP	87.4 MB	67.2 MB	24661 ms	Exitoso	Exitoso
		REST	87.4 MB	67.2 MB	22784 ms	Exitoso	Exitoso
32	78.2 MB	SOAP	101.0 MB	78.2 MB	25147 ms	Exitoso	Exitoso
		REST	101.0 MB	78.2 MB	24991 ms	Exitoso	Exitoso
33	95.6 MB	SOAP	124.0 MB	95.6 MB	28473 ms	Exitoso	Exitoso
		REST	124.0 MB	95.6 MB	26108 ms	Exitoso	Exitoso
34	98.8 MB	SOAP	128.0 MB	98.8 MB	28967 ms	Exitoso	Exitoso
		REST	128.0 MB	98.8 MB	26531 ms	Exitoso	Exitoso
35	133.5 MB	SOAP	173.0 MB	133.5 MB	32188 ms	Exitoso	Exitoso
		REST	173.0 MB	133.5 MB	28019 ms	Exitoso	Exitoso
36	293.4 MB	SOAP	382.0 MB	293.4 MB	42078 ms	Exitoso	Exitoso
		REST	382.0 MB	293.4 MB	40071 ms	Exitoso	Exitoso
37	314.0 MB	SOAP	409.0 MB	314.0 MB	46109 ms	Exitoso	Exitoso
		REST	409.0 MB	314.0 MB	44374 ms	Exitoso	Exitoso
38	409.9 MB	SOAP	546.0 MB	409.9 MB	106532 ms	Exitoso	Exitoso
		REST	546.0 MB	409.9 MB	98082 ms	Exitoso	Exitoso
39	528.3 MB	SOAP	688.0 MB	528.3 MB	128307 ms	Exitoso	Exitoso
		REST	688.0 MB	528.3 MB	121549 ms	Exitoso	Exitoso

Los recursos de aprendizaje reconstruidos, tanto para SOAP como para REST, no presentaron daños en sus contenidos y se entregaron ocupando el mismo tamaño de memoria del recurso original antes de ser segmentado y empaquetado como un Servicio Web de Aprendizaje. Por lo que, se verifica que el recurso de aprendizaje se reconstruye de manera

correcta. Los resultados variaron únicamente en los tiempos de reconstrucción, siendo con la arquitectura REST donde se obtuvieron un menor tiempo de respuesta, aunque cabe mencionar que los tiempos pueden variar dependiendo de las configuraciones del equipo y del entorno de pruebas.

3^{er} Tipo de Prueba: “Composición de Servicios Web de Aprendizaje con contenidos segmentados”

Esta prueba consiste en la composición de al menos dos Servicios Web de Aprendizaje generados mediante segmentación, tanto para SOAP como para REST. Los datos utilizados en esta prueba son indicados en la Tabla C.5.

Tabla C.5. Datos utilizados en el Tipo de Prueba 3

Nombre de la Prueba	Composición de Servicios Web de Aprendizaje con contenidos segmentados
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta composición de Servicios Web de Aprendizaje Compuestos con contenidos de gran tamaño que han sido segmentados mediante SOAP y REST.
Precondiciones	<ul style="list-style-type: none"> ▪ Los campos en el formulario deben ser llenados y válidos. ▪ El componente de software debe funcionar correctamente. ▪ Deben existir al menos dos servicios web de aprendizaje y estar alojados en un servidor de aplicaciones.
Entrada	Dos o más Servicios Web de Aprendizaje que han sido generados mediante segmentación y se encuentren alojados en un servidor de aplicaciones.
Salida	<ul style="list-style-type: none"> ▪ La generación del archivo Java correspondiente al nuevo Servicio Web de Aprendizaje Compuesto, el cual contiene el arreglo donde se invocan a todos los servicios web de aprendizaje ingresados.
Criterio de Éxito	<p>Cuando el Servicio Web de Aprendizaje Compuesto se genera de manera correcta, se entregará la siguiente notificación en la vista de la aplicación web:</p> <div style="border: 1px solid #ccc; background-color: #e6f2e6; padding: 5px; width: fit-content; margin: 10px auto;"> LWS composite created Close </div> <p>De caso contrario, nos informará que ocurrió un error al momento de la composición.</p>

Para la composición de Servicios Web de Aprendizaje Compuestos se consideró tomar un dominio y tema de aprendizaje en específico cuyos contenidos permitieran llevar un seguimiento desde el objetivo al contenido y de ahí pasar a las actividades y/o evaluaciones de ser necesarias.

Se consideraron otros recursos de aprendizaje para completar el curso del tema de aprendizaje, por lo que se generaron nuevos Servicios Web de Aprendizaje para estos, mismos que se crearon para SOAP y REST. Aunque el atributo más relevante a considerar es el tamaño del recurso. Se realizaron las pruebas de composición, las cuales se presentan en la Tabla C.6.

Tabla C.6. Resultados de composición para SWA con contenidos segmentados

Nombre del SWA	Dominio y Tema de Aprendizaje	Tamaño del SWA Contenido	Formato	Tipo	Protocolos Generados	Respuesta de Composición
Piano_Obj	Música: "Piano"	0.32 MB	PNG	Objetivo	SOAP y REST	Exitosos
Piano_Cont		22.1 MB	PDF	Contenido		
Piano_Act		101.0 MB	MP4	Actividad		
Alcanos_Obj	Química: "Alcanos, alquenos y alquinos"	0.45 MB	PDF	Objetivo	SOAP y REST	Exitosos
Alcanos_Cont		173.0 MB	MP4	Contenido		
Alcanos_Act		5.19 MB	PDF	Actividad		
Alcanos_Eval		0.69 MB	PDF	Evaluación		
Prog_Cont	Computación: "Programación"	14.8 MB	PDF	Contenido	SOAP y REST	Exitosos
Prog_Cont2		124.0 MB	PDF	Contenido		
Prog_Cont3		382.0 MB	PDF	Contenido		
Algeb_Cont	Matemáticas: "Álgebra"	42.4 MB	MP4	Contenido	SOAP y REST	Exitosos
Algeb_Cont2		409.0 MB	PDF	Contenido		
Algeb_Cont3		688.0 MB	MP4	Contenido		

El resultado en cada prueba de composición fue exitoso demostrando que la herramienta es capaz de realizar composición para Servicios Web de Aprendizaje con contenidos de gran tamaño que han sido previamente segmentados, e incluso con recursos pequeños que no fueron segmentados.

La clase java resultante a esta composición reconocerá a cada SWA invocado en ella de acuerdo a su protocolo de transferencia, esto porque en el caso de SOAP la URL para invocar al servicio tiene la terminación "Impl?wsdl", mientras que si el servicio es de tipo REST no la tendrá. Por lo que, se importarán solo las librerías y métodos necesarios en el archivo del Servicio Web de Aprendizaje Compuesto correspondientes a cada protocolo.

Los Servicios Web de Aprendizaje Compuestos que se generaron y los servicios a los que invocan se encuentran listados en la Tabla C.7.

Tabla C.7. Lista de SWA-C generados en SOAP y REST

Nombre del SWA-C	Protocolo	SWA Invocado	Tipo	Tamaño Acumulado
Piano_Co_SOAP	SOAP	Piano_Obj_SOAP	Objetivo	143.42 MB
		Piano_Cont_SOAP	Contenido	
		Piano_Act_SOAP	Actividad	
Piano_Co_REST	REST	Piano_Obj_REST	Objetivo	143.42 MB
		Piano_Cont_REST	Contenido	
		Piano_Act_REST	Actividad	
Alcanos_Co_SOAP	SOAP	Alcanos_Obj_SOAP	Objetivo	179.33 MB
		Alcanos_Cont_SOAP	Contenido	
		Alcanos_Act_SOAP	Actividad	
		Alcanos_Eval_SOAP	Evaluación	
Alcanos_Co_REST	REST	Alcanos_Obj_REST	Objetivo	179.33 MB
		Alcanos_Cont_REST	Contenido	
		Alcanos_Act_REST	Actividad	
		Alcanos_Eval_REST	Evaluación	
Program_Co_SOAP	SOAP	Prog_Cont_SOAP	Contenido	520.80 MB
		Prog_Cont2_SOAP	Contenido	

		Prog_Cont3_SOAP	Contenido	
Program_Co_REST	REST	Prog_Cont_REST	Contenido	520.80 MB
		Prog_Cont2_REST	Contenido	
		Prog_Cont3_REST	Contenido	
Algebra_Co_SOAP	SOAP	Algeb_Cont_SOAP	Contenido	1,521.40 MB
		Algeb_Cont2_SOAP	Contenido	
		Algeb_Cont3_SOAP	Contenido	
Algebra_Co_REST	REST	Algeb_Cont_REST	Contenido	1,521.40 MB
		Algeb_Cont2_REST	Contenido	
		Algeb_Cont3_REST	Contenido	

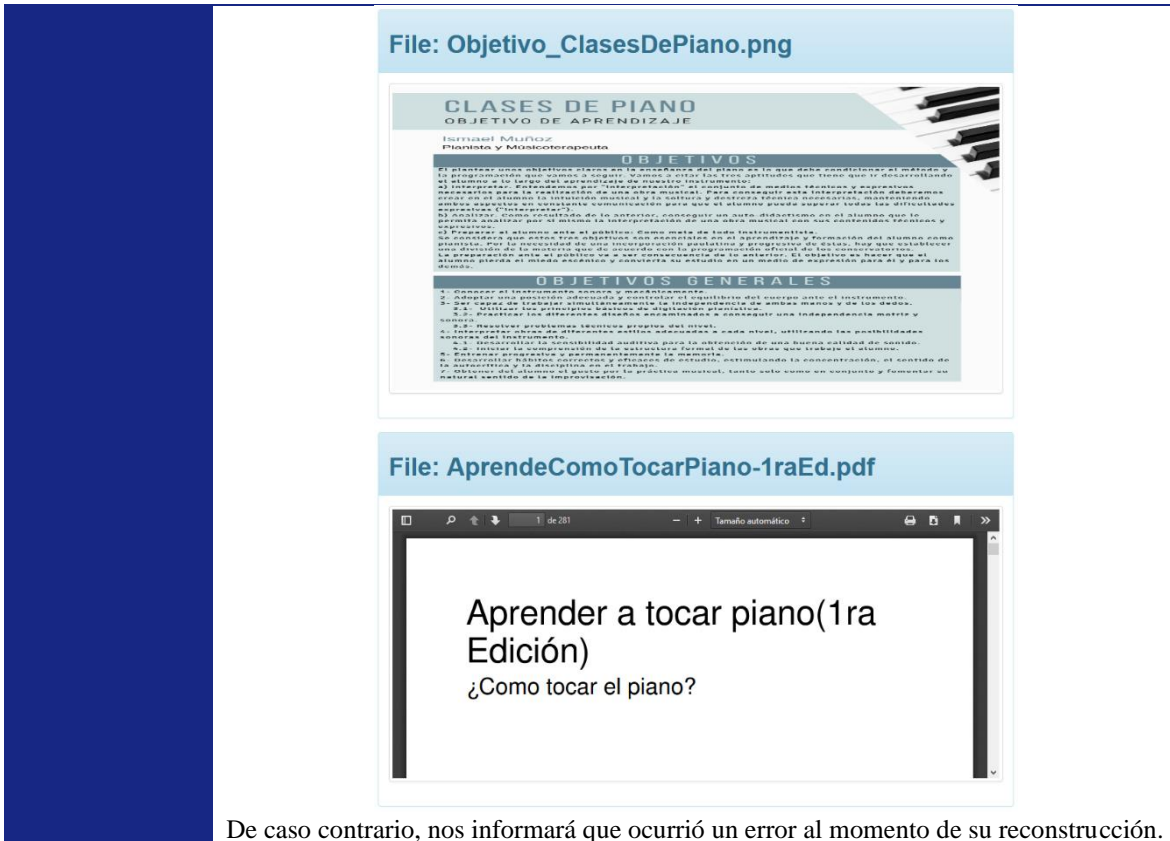
Cabe destacar que se utilizaron mayormente recursos con objetos de aprendizaje de tipo contenido, debido a que estos son los que comúnmente tienen mayor tamaño y resultan idóneos para realizar las pruebas de segmentación, reconstrucción, presentación y composición. Siendo el atributo del tamaño, el principal en este trabajo de investigación.

4^{to} Tipo de Prueba: “Presentación de Servicios Web de Aprendizaje Compuestos con contenidos segmentados”

Esta prueba consiste en la correcta presentación de los Servicios Web de Aprendizaje Compuestos que fueron generados en el Tipo de Prueba 3 para SOAP y REST. Para la realización de esta prueba se utilizan los datos indicados en la Tabla C.8.

Tabla C.8. Datos utilizados en el Tipo de Prueba 4

Nombre de la Prueba	Presentación de Servicios Web de Aprendizaje Compuestos con contenidos segmentados
Ejecutor	Iván Humberto Fuentes Chab
Objetivo	Verificar la correcta reconstrucción de todos los servicios contenidos en un Servicio Web de Aprendizaje Compuesto, para que tengan una adecuada presentación al cliente a través de SOAP y REST.
Precondiciones	<ul style="list-style-type: none"> ▪ El campo en el formulario debe ser llenado y válido. ▪ El componente de software debe funcionar correctamente. ▪ El servicio web de aprendizaje compuesto y todos sus servicios atómicos invocados deben estar alojados en servidor.
Entrada	Uno de los Servicios Web de Aprendizaje Compuestos enlistados en la Tabla C.7 que fueron generados en el Tipo de Prueba 3.
Salida	Los recursos contenidos por el Servicio Web de Aprendizaje Compuesto serán reconstruidos y presentados en la vista del Cliente PHP.
Criterio de Éxito	Cuando los recursos del Servicio Web de Aprendizaje Compuesto son reconstruidos de manera correcta, se presentarán todos los recursos contenidos en la vista del cliente web:



De caso contrario, nos informará que ocurrió un error al momento de su reconstrucción.

Cuando se trabajan con recursos de gran tamaño, el servidor tiende a emplear todo el esfuerzo en su reconstrucción para poder presentarlos. Es por tal motivo que deben tenerse por separado los servicios web de aprendizaje cuando se alojan en un servidor, realizando composiciones mediante separación de protocolos de transferencia como se sugiere en la guía de composición de este trabajo.

Por tal motivo se desarrollaron dos vistas en el cliente PHP, uno correspondiente a cada protocolo para evitar sobrecarga de memoria, ya que si se tiene un proyecto que contenga servicios tanto en SOAP como en REST generará las librerías y métodos requeridos para cada uno, y al estar trabajando con recursos contenidos de gran tamaño, afectará al momento de cargarlos lo que ocasionaría que estos recursos no lleguen a presentarse.

Los resultados obtenidos para la presentación de los Servicios Web de Aprendizaje Compuestos con contenidos, se presentan en la Tabla C.9.

Tabla C.9. Resultados de presentación para SWA-C con contenidos segmentados

Nombre del SWA-C	Cantidad de Invocaciones	Tamaño Acumulado	Tiempo de Presentación	Respuesta de Presentación
Piano_Co_SOAP	3 SWA en SOAP	143.42 MB	24782 ms	Exitoso
Piano_Co_REST	3 SWA en REST	143.42 MB	21604 ms	Exitoso
Alcanos_Co_SOAP	4 SWA en SOAP	179.33 MB	32540 ms	Exitoso
Alcanos_Co_REST	4 SWA en REST	179.33 MB	32687 ms	Exitoso
Program_Co_SOAP	3 SWA en SOAP	520.80 MB	126370 ms	Exitoso

Program_Co_Rest	3 SWA en REST	520.80 MB	98546 ms	Exitoso
Algebra_Co_SOAP	3 SWA en SOAP	1,521.40 MB	402815 ms	Exitoso
Algebra_Co_REST	3 SWA en REST	1,521.40 MB	346590 ms	Exitoso

Las pruebas realizadas para presentar Servicios Web de Aprendizaje Compuestos resultaron exitosas, donde se lograron visualizar cada uno de sus recursos contenidos sin verse dañados o afectados por su infraestructura. Se considera al tiempo de presentación en esta prueba, como al lapso de tiempo tomado por el cliente PHP para mostrar los recursos en la vista del navegador, donde en cada prueba realizada se llegaron a visualizar todos sus recursos.

Anexo

D

Descripción de las Pruebas Adicionales

Este anexo presenta las Pruebas Adicionales realizadas de manera más detallada, así como los resultados obtenidos en cada una de ellas.

Comportamiento de los Recursos de Aprendizaje en distintos formatos

De los 39 recursos de aprendizaje enlistados en la Tabla 5.2 en el Capítulo 5, se clasifican por sus tipos de formatos para observar sus comportamientos en los procesos de segmentación y reconstrucción. Ubicando a los recursos 10, 13, 27, 32, 35, 38 y 39 de tipo MP4 como se aprecia en la Figura D.1; a los recursos 9, 11, 12, 14, 15, 16, 17, 20, 22, 33, 34, 36 y 37 de tipo PDF como se aprecia en la Figura D.2; a los recursos 19, 23, 25, 28 y 30 de tipo DOC como se aprecia en la Figura D.3; y a los recursos 18, 21, 24, 29 y 31 de tipo PPT como se aprecia en la Figura D.4.

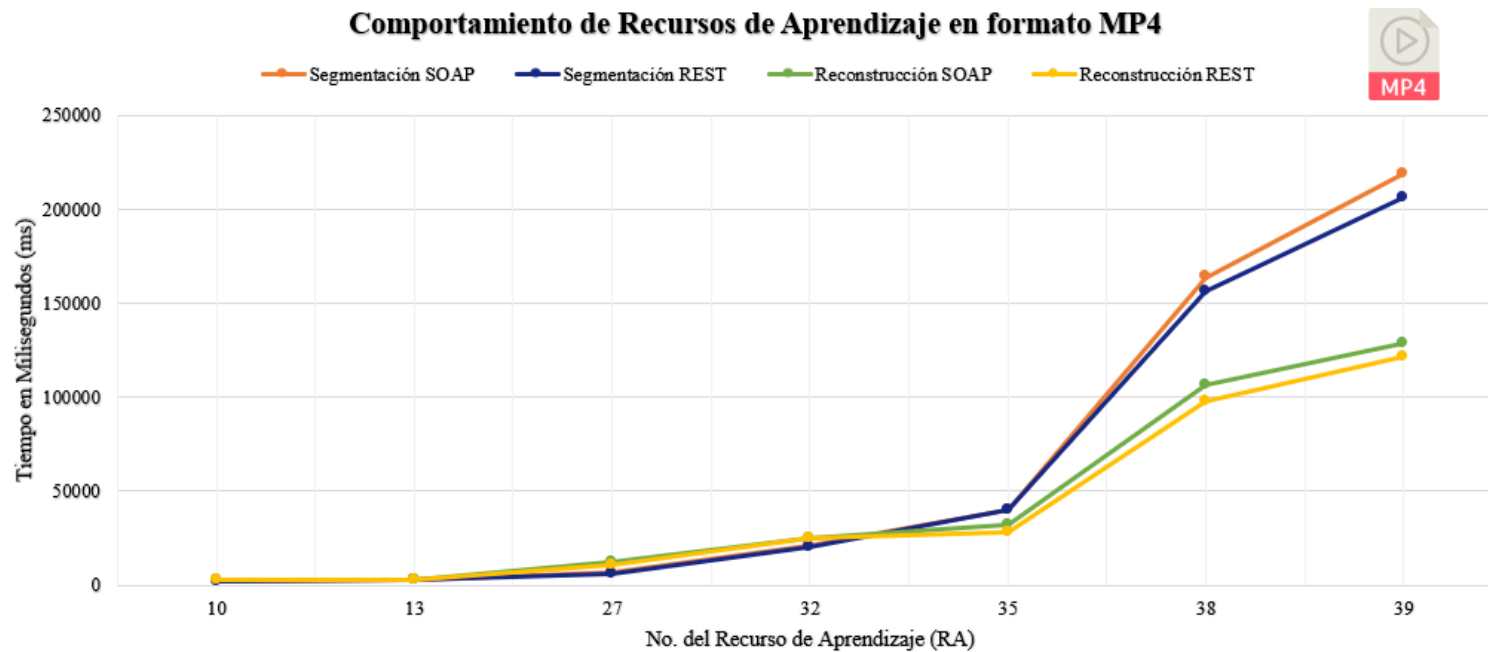


Figura D.1. Comportamiento de RA en formato MP4

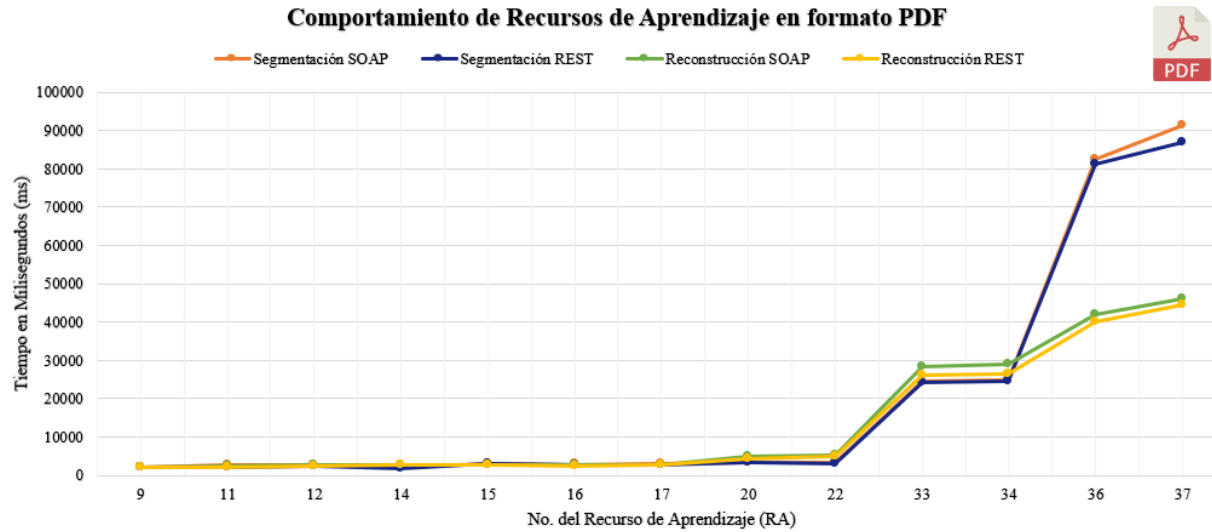


Figura D.2. Comportamiento de RA en formato PDF

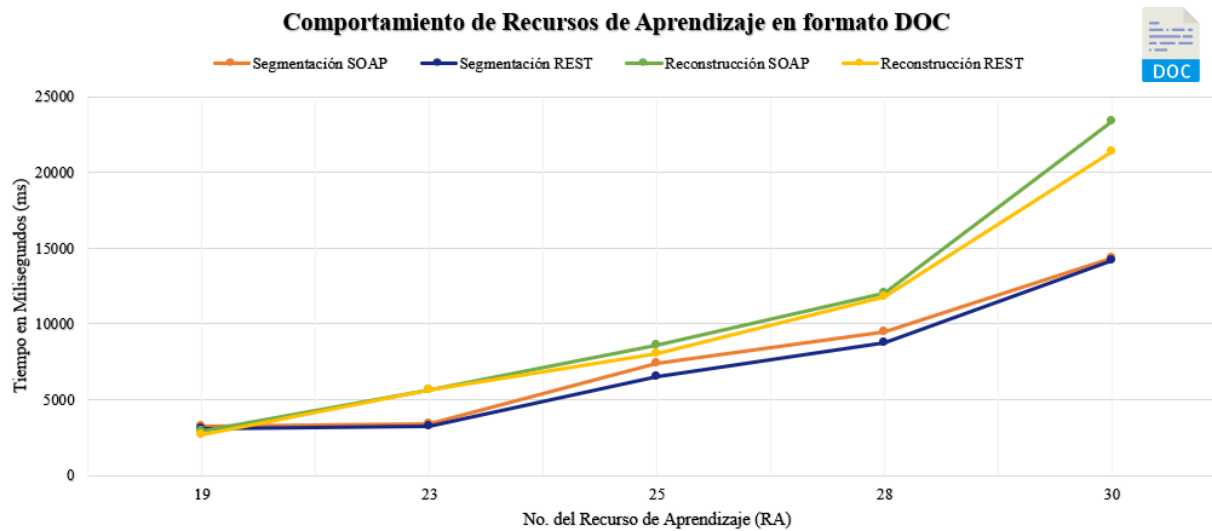


Figura D.3. Comportamiento de RA en formato DOC

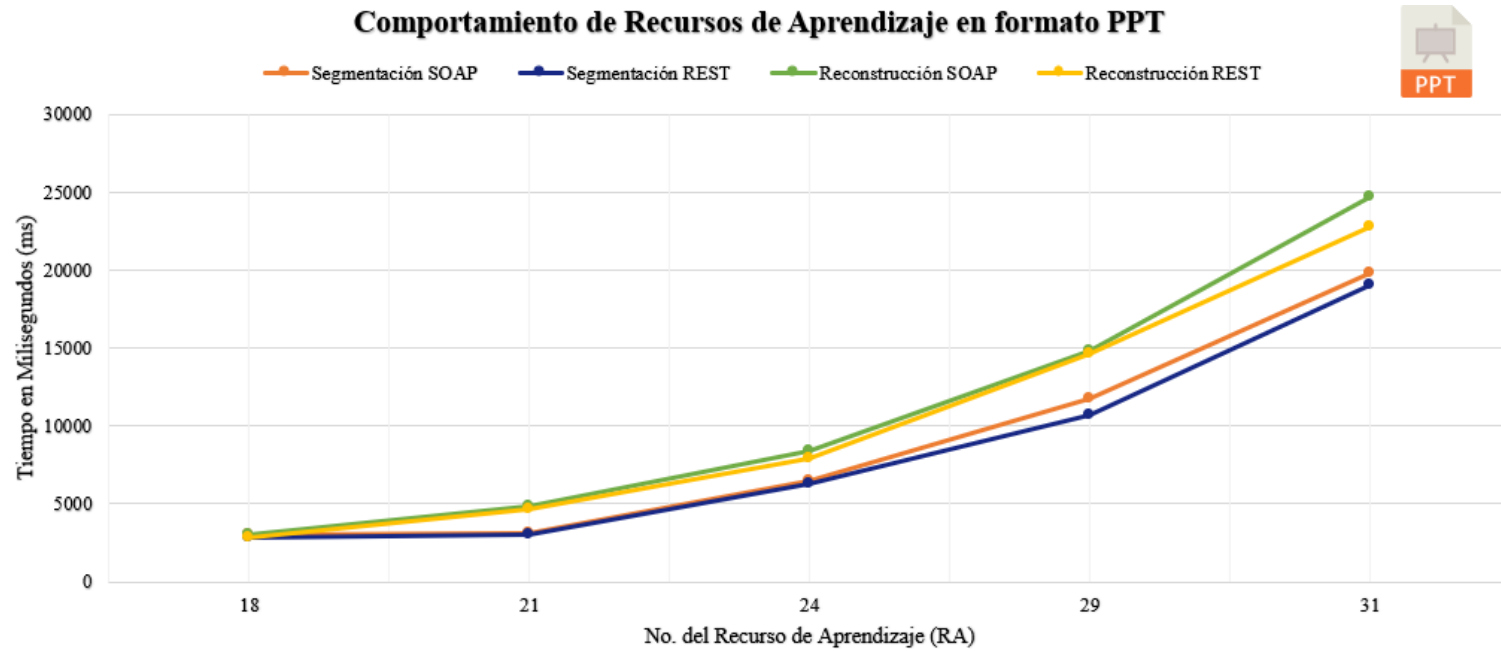


Figura D.4. Comportamiento de RA en formato PPT

Los comportamientos de los Recursos de Aprendizaje para los distintos formatos indican que, a mayor tamaño del recurso, mayores serán los tiempos de respuesta. En cada uno de las gráficas se comparan los tiempos promedio para los procesos de segmentación y reconstrucción, donde puede observarse que no existe alteración alguna entre tipos de formatos.