



Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativos

Tesista: **Lic. Alejandro Sartorio**
Director: **Dr. Gustavo Rossi**

TESIS DOCTORAL

Doctorado en Tecnología Informática
Facultad de Tecnología Informática

Rosario, Argentina
2026

Miembros del Jurado:

- Dr. Alfredo Capozucca *University of Luxembourg*
- Dr. Guillermo Rodriguez *Universidad Nacional de Rosario*
- Dr. Battaglia Nicolas *Universida Abierta Interamericana*

Resumen

La educación digital actual enfrenta desafíos estructurales derivados de la heterogeneidad de los perfiles estudiantiles. Si bien los *Learning Management Systems* (LMS) son la infraestructura predominante, su rigidez arquitectónica limita la personalización a escala. El problema central identificado en esta investigación no reside en la carencia de algoritmos, sino en la ausencia de arquitecturas de referencia que permitan una transición evolutiva y no intrusiva desde sistemas legados hacia Sistemas de Aprendizaje Adaptativo (ALS).

Como contribución principal, se presenta ARALS (*Architecture of Reference for Adaptive Learning Systems*), un artefacto puente interdisciplinario diseñado para traducir principios pedagógicos en decisiones arquitectónicas verificables. Estructurada en seis capas funcionales y fundamentada en el ciclo de control autónomo MAPE-K, ARALS permite desacoplar la lógica de adaptación de las funciones básicas del LMS, garantizando atributos de calidad como escalabilidad e interoperabilidad mediante estándares internacionales (LTI, xAPI).

Metodológicamente, la tesis introduce el marco de propiedades PALS-A, destacando la Propiedad PAC (Personalización Adaptativa Colaborativa) como el núcleo que integra la adaptación continua con la colaboración contextual. Asimismo, se define una hoja de ruta de transformación modelada en BPMN para guiar la refactorización sistemática de plataformas existentes hacia entornos adaptativos.

La viabilidad de la propuesta se validó empíricamente en la plataforma Odoo eLearning. El proceso demostró la capacidad de inyectar funciones adaptativas en entornos de producción sin comprometer la estabilidad operativa. La evaluación, complementada con el método ATAM y estudios de usabilidad, confirmó que ARALS reduce significativamente la brecha de comunicación entre perfiles pedagógicos y técnicos. En conclusión, este trabajo consolida un marco operativo sistemático y reproducible donde la arquitectura de software actúa como el garante de la calidad pedagógica.

RESUMEN

Palabras clave: Arquitectura de Referencia, ARALS, Sistemas de Aprendizaje Adaptativo (ALS), MAPE-K, PALS-A, Transformación Arquitectónica, Interoperabilidad Educativa.

Agradecimientos y dedicatorias

Al finalizar esta etapa fundamental de mi formación doctoral, quiero expresar mi más profundo agradecimiento a todas las personas e instituciones que hicieron posible el desarrollo de esta investigación.

En primer lugar, al **Dr. Gustavo Rossi**, director de esta tesis, por su generosidad, guía académica y acompañamiento constante durante este proceso.

Al **Dr. Marcelo De Vicenzi** y al **Dr. Carlos Neil**, por el apoyo institucional, la confianza y por abrir los caminos necesarios para que este trabajo llegara a buen puerto.

A la **Dra. Patricia San Martín**, por su aporte fundamental en mi formación inicial en investigación y por su valiosa visión sobre la relación entre educación y tecnología.

A mis colegas y docentes de la **Universidad Abierta Interamericana (UAI)**, por el acompañamiento y el intercambio académico a lo largo de este recorrido.

Un agradecimiento especial a mis socios de Hitofusion, **Ing. Matias Banega** y **Ing. Lucas Burgos**, por el apoyo permanente y por sostener conmigo cada desafío profesional y humano durante esta etapa.

A mis **amigos**, por ser el cable a tierra necesario, por los momentos compartidos y por recordarme siempre el valor de las raíces.

A mi familia. A mi madre **Rosa**, a mi padre **Roberto**, a mi abuela **Ángela** y a mi hermano **Omar**, por el amor, el esfuerzo y los valores que me acompañan en cada paso.

Finalmente, y de manera muy especial, a mi esposa **Lara Capponi**. Gracias por ser la mejor compañera en cada desafío y por estar a mi lado en cada momento de estudio. Esta tesis es también fruto de tu apoyo constante.

Índice general

Resumen	III
Índice de tablas	X
Índice de figuras	XI
1. Introducción	1
1.1. Contexto y Motivación	1
1.2. Planteamiento del Problema	3
1.3. Objetivos de la Investigación	9
1.3.1. Objetivo General	9
1.3.2. Objetivos Específicos	10
1.4. Contribución	12
1.4.1. Trabajos Presentados Vinculados con la Tesis	16
1.5. Estructura de la Tesis	17
2. Estado del Arte y Fundamentación Teórica	19
2.1. Introducción	19
2.2. Marco Teórico	20
2.2.1. Fundamentos de Arquitectura de Software	20
2.2.2. Bases Conceptuales de Sistemas de Aprendizaje Adaptativo . .	21
2.2.3. Teorías de Adaptación en Sistemas Computacionales	22
2.2.4. Principios de Diseño Arquitectónico para la Adaptabilidad . .	23
2.2.5. Marcos de Referencia Conceptuales	23
2.3. Estado del Arte: Análisis de la Literatura	24
2.3.1. Metodología de Selección y Análisis de Fuentes	25
2.3.2. Análisis de Arquitecturas Actuales	28

2.3.3.	Tecnologías y Patrones Emergentes	29
2.3.4.	Casos de Estudio Representativos	30
2.3.5.	Síntesis de Hallazgos y Brechas Identificadas	30
2.4.	Trabajos Relacionados	31
2.4.1.	El Problema Central y su Evolución en el Estado del Arte . . .	32
2.4.2.	Metodología de Análisis Comparativo	32
2.4.3.	Análisis de Propuestas y Síntesis de Hallazgos	33
2.4.4.	Identificación de Brechas y Oportunidades: Hacia ARALS . . .	33
2.5.	Fundamentos para la Construcción de la Arquitectura de Referencia .	34
2.5.1.	Contribuciones Teóricas	35
2.5.2.	Hallazgos Empíricos	35
2.5.3.	Direcciones Futuras	36
3.	Fundamentos Metodológicos para la Transformación Arquitectónica hacia ARALS	38
3.1.	Introducción	38
3.2.	Fase 1: Caracterización de arquitecturas LMS	39
3.2.1.	Núcleo Funcional y Estructuras Invariantes	41
3.2.2.	Análisis de Patrones y Estilos Arquitectónicos	41
3.2.3.	Implicaciones para el Diseño de ARALS	44
3.3.	Fase 2: Generalización Arquitectónica	44
3.3.1.	Estructura Estratificada y Relaciones de Control	44
3.3.2.	El Modelo RA-LMS: Invariantes y Extensibilidad	46
3.4.	Relación entre propiedades, componentes y arquitectura subyacente .	48
3.5.	Fase 3: Incorporación de capacidades adaptativas	49
3.5.1.	Propiedades Adaptativas PALS-A: Atributos Arquitectónicamente Influentes	50
3.5.2.	Mapeo de Propiedades y Estilos Arquitectónicos	51
3.6.	Fase 4: Arquitectura General Unificada	51
3.6.1.	Estructura Integrada por Capas y Estilos	51
3.6.2.	Síntesis y Transición hacia ARALS	52
4.	Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo	56
4.1.	Introducción	56
4.2.	Diseño de la arquitectura de referencia	58
4.2.1.	Estructura general y principios de diseño	62

4.2.2. Capa de Presentación – Interfaces y Experiencia del Usuario . . .	63
4.2.3. Capa de Aplicación y Servicios – Núcleo Funcional del Sistema . . .	64
4.2.4. Capa de Lógica de Negocio – Núcleo de Procesos y Reglas . . .	66
4.2.5. Capa de Integración – Conectividad y Estándares Educativos . . .	67
4.2.6. Capa de Seguridad – Protección, Control y Cumplimiento . . .	69
4.2.7. Capa de Datos – Almacenamiento, Persistencia y Rendimiento . . .	71
4.2.8. Capa de Servicios Externos – Integraciones y Soportes Complementarios	72
4.3. Validación Estructural y Funcional del Modelo ARALS	74
5. Hoja de Ruta para la Promoción de Propiedades Adaptativas en Sistemas de Aprendizajes	81
5.1. Introducción	81
Introducción	81
5.2. Propiedades Adaptativas en Sistemas de Aprendizaje	83
5.3. Guía de transformación LMS→ALS	86
5.4. Etapa 1 — Identificación de Propiedades Adaptativas y Definición del Punto de Partida de la Transformación	88
5.5. Etapa 2 — Diagnóstico de la Arquitectura <i>as-is</i>	90
5.6. Etapa 3 — Ejecución de la Transformación Arquitectónica	92
5.7. Etapa 4 — Validación y Aseguramiento de la Calidad Adaptativa . . .	95
6. Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning	97
6.1. Introducción	97
6.2. Caracterización de los Requerimientos de Mejoras para Odoo Elearning . . .	99
6.3. Aplicación Etapa 1 — Identificación de Propiedades Adaptativas . . .	100
6.4. Aplicación Etapa 2 – Análisis de la Arquitectura <i>as-is</i>	102
6.5. Aplicación Etapa 3 – Transformación Arquitectónica	105
6.6. Aplicación Etapa 4 – Validación de la Transformación Arquitectónica . . .	108
6.7. Detalles de la implementación técnica y estructura de componentes . .	109
6.7.1. Implementación de Patrones y Componentes Críticos	111
6.8. Contribuciones funcionales para la Experiencia de Usuario del caso de uso	113

7. Discusión, Contribuciones y Conclusiones	119
7.1. Discusión de Resultados	119
7.2. Sistematización de Contribuciones	120
7.3. Limitaciones del Trabajo	121
7.4. Líneas Futuras de Investigación	122
7.5. Reflexión Final	123
Bibliografía	125
Glosario	137

Índice de tablas

2.1. Matriz de Evaluación Comparativa de Trabajos Relacionados	33
3.1. Síntesis de patrones arquitectónicos y funcionalidades en LMS	41
3.2. Matriz de componentes arquitectónicos identificados en plataformas LMS	42
3.3. Matriz de trazabilidad: Componentes RA-LMS, Estilos y Propiedades PALS.	48
3.4. Relación entre propiedades PALS-A y estilos arquitectónicos habilitadores.	51
4.1. Matriz de relaciones entre capas (Rel1–Rel13, versión compacta)	59
4.2. Relación entre capas, patrones arquitectónicos y tecnologías asociadas en ARALS	75
4.3. Validación de cobertura de componentes en arquitectura de referencia ARALS	78
4.4. Validación de estilos arquitectónicos en ARALS	79
5.1. Ficha técnica de la Etapa 2: Diagnóstico Arquitectónico.	90
5.2. Ficha técnica de la Etapa 3: Transformación Arquitectónica.	93
5.3. Ficha técnica de la Etapa 4: Validación y Aseguramiento.	95
6.1. Correspondencia entre requerimientos y propiedades PALS	101
6.2. Caso de uso previo a la implementación de rutas adaptativas	102
6.3. Casos de uso de rutas de aprendizaje para el módulo modificado a través de la arquitectura ARALS	103
6.4. Comparativa Arquitectura Modelo y Odoo eLearning	104
6.5. Comparación de materiales didácticos entre Odoo eLearning estándar y versión ARALS	114

Índice de figuras

3.1. Fases del proceso analítico y preparatorio para la construcción de la Arquitectura de Referencia ARALS.	40
3.2. Modelo estratificado del dominio LMS: Organización de componentes funcionales e identificación de los puntos de integración para servicios transversales y adaptativos.	43
3.3. Abstracción de capas y dominios funcionales.	45
3.4. Vista relacional: los conectores como elementos de primera clase. . . .	47
3.5. Arquitectura de Referencia RA-LMS: Modelo canónico diseñado bajo principios de DDD y el patrón Microkernel para garantizar la extensibilidad no intrusiva.	54
3.6. Arquitectura General Unificada (ARALS-Preliminar): Síntesis estructural que articula el núcleo transaccional con los metatipos adaptativos T1–T5 para la inyección del ciclo MAPE-K.	55
4.1. Arquitectura de Referencia ARALS: Representación estratificada en seis capas funcionales y su red de trece relaciones semánticas (Rel1–Rel13) para la gestión del ciclo adaptativo.	61
4.2. Capa de Presentación: Organización de componentes para la orquestación de la experiencia de usuario y la personalización dinámica de la interfaz mediante el <i>Adaptive UI Engine</i>	63
4.3. Capa de Aplicación y Servicios: Centro de orquestación operativa que integra servicios educativos desacoplados y servicios de aprendizaje adaptativo bajo un enfoque SOA.	65
4.4. Capa de Lógica de Negocio: Núcleo cognitivo del sistema donde residen los motores de inferencia, el modelado del estudiante y la gestión de trayectorias de aprendizaje dinámicas.	66

4.5. Capa de Integración: Subsistema de interoperabilidad encargado de la conectividad con herramientas externas y la gestión de estándares educativos internacionales (LTI, SCORM, xAPI, QTI).	68
4.6. Capa de Seguridad: Marco transversal para la gobernanza de datos, el cumplimiento normativo (GDPR) y el aseguramiento de la confianza ética en los procesos adaptativos.	70
4.7. Capa de Datos: Infraestructura de persistencia políglota para el almacenamiento de perfiles de usuario, objetos de aprendizaje y trazabilidad de eventos analíticos en el LRS.	71
4.8. Capa de Servicios Externos: Vector de expansión tecnológica que habilita la colaboración con infraestructuras en la nube, servicios de IA/ML y plataformas de analítica externas.	73
4.9. Validación Estructural MAPE-K: Correspondencia entre las capas funcionales de la arquitectura ARALS y las fases del ciclo de control autónomico (<i>Monitor, Analyze, Plan, Execute, Knowledge</i>).	76
5.1. Flujo de trabajo de la Hoja de Ruta de transición LMS – > ALS	87
6.1. Arquitectura <i>as-is</i> del módulo eLearning de Odoo: Punto de partida para el Diagnóstico de Deuda Técnica y análisis de brechas estructurales (Etapa 2)	106
6.2. Arquitectura <i>to-be</i> (Modelo C4): Incremento de Software Adaptativo mediante la inyección de la Capa de Servicios y desacoplamiento de componentes (Etapa 3).	107
6.3. Organización de componentes en la arquitectura base (<i>as-is</i>): Evidencia de la rigidez estructural previa a la transformación	110
6.4. Materialización del desacoplamiento funcional y la creación de la capa de servicios adaptativos (Etapa 3).	111
6.5. Interfaces de configuración adaptativa: Instrumentación funcional de la propiedad PAC y el motor de perfiles de usuario (Etapa 4)	115
6.6. Interfaz de navegación extendida: Punto de acceso a los servicios adaptativos e instrumentación de las propiedades PALS en el frontend (Etapa 3)	116
6.7. Validación funcional de rutas adaptativas: Ejecución de la secuenciación dinámica basada en el desempeño del aprendiz (Etapa 4)	117
6.8. Visualización de métricas de progreso: Verificación de la Propiedad de Retroalimentación Adaptativa y monitoreo continuo (Etapa 4)	117

6.9. Interfaz adaptativa del estudiante: Validación de la Personalización Adaptativa Colaborativa (PAC) y ejecución de rutas dinámicas (Etapa 4) .	118
6.10. Tablero analítico para el docente: Operacionalización de la Propiedad Multistakeholder y transparencia algorítmica (Etapa 4)	118

Capítulo 1

Introducción

1.1. Contexto y Motivación

La educación contemporánea enfrenta desafíos sin precedentes derivados de la masificación del acceso, la heterogeneidad creciente de los perfiles de estudiantes, las expectativas de personalización del aprendizaje, y la necesidad de garantizar equidad y calidad en contextos de recursos limitados [1, 2]. Los *Learning Management Systems* (LMS) —plataformas tecnológicas que gestionan contenidos, actividades, evaluaciones y comunicación en entornos educativos— constituyen la infraestructura digital dominante en instituciones de educación superior, educación a distancia y formación corporativa [3, 4]. Sin embargo, estos sistemas fueron concebidos originalmente como herramientas de gestión transaccional centradas en la administración de recursos y el seguimiento de actividades, sin capacidades sofisticadas de personalización, adaptación dinámica ni inteligencia contextual [5, 6].

Paralelamente, la investigación en *Adaptive Learning Systems* (ALS) ha demostrado que la personalización de rutas de aprendizaje, la adaptación de contenidos al perfil cognitivo del estudiante, la retroalimentación contextualizada y el ajuste dinámico de estrategias pedagógicas pueden mejorar significativamente los resultados educativos, reducir tasas de abandono y promover experiencias de aprendizaje más efectivas y motivadoras [7, 8]. No obstante, la mayoría de las propuestas de sistemas adaptativos documentadas en la literatura científica corresponden a prototipos de investigación, soluciones ad-hoc para dominios específicos, o extensiones puntuales de plataformas comerciales, sin constituir arquitecturas sistemáticas, escalables, interoperables y transferibles que puedan adoptarse de manera sostenible en instituciones educativas reales

[9, 10].

Esta brecha entre la potencialidad teórica de los ALS y su adopción práctica en contextos institucionales se agrava por una **desconexión estructural entre las disciplinas que intervienen en el diseño, despliegue y operación de tecnología educativa**. Por un lado, los *expertos en pedagogía* conceptualizan el aprendizaje desde marcos cognitivos, constructivistas y socioculturales, enfatizando la autorregulación, la mediación social y el andamiaje [11, 12, 13]. Estas perspectivas buscan promover procesos de enseñanza-aprendizaje que reconozcan la complejidad contextual del aprendizaje humano y su carácter dinámico.

Por otro lado, los *analistas funcionales de plataformas educativas* traducen requerimientos pedagógicos en especificaciones técnicas, aunque frecuentemente sin una comprensión profunda de los modelos teóricos que sustentan el aprendizaje adaptativo. Esta brecha ha sido señalada de manera recurrente en la literatura sobre sistemas de aprendizaje adaptativos, que advierte la necesidad de integrar marcos conceptuales sólidos con mecanismos computacionales formales [7, 6]. La desconexión se amplifica cuando se consideran los desafíos que plantea el modelado del estudiante y la personalización basada en datos [8, 14].

A su vez, los *desarrolladores de software* implementan soluciones bajo criterios de eficiencia, modularidad, patrones arquitectónicos y atributos de calidad [15, 16]. Sin embargo, estas decisiones suelen tomarse sin una articulación explícita con teorías del aprendizaje o principios pedagógicos, generando plataformas que funcionan correctamente desde el punto de vista computacional pero que no necesariamente responden de manera adecuada a la complejidad educativa. Estudios recientes sobre arquitecturas para sistemas adaptativos remarcan que, sin un puente conceptual entre disciplinas, las soluciones tecnológicas tienden a fragmentarse y a perder coherencia sistémica [17, 18].

Finalmente, los implementadores provenientes del ecosistema ERP (como Odoo, SAP o Microsoft Dynamics) y de LMS tradicionales (como Moodle, Open edX, Sakai, Chamilo o Blackboard) suelen abordar las plataformas educativas desde lógicas transaccionales y operativas [3, 4]. Este enfoque privilegia flujos administrativos, gestión de contenidos y operaciones estandarizables, lo cual resulta expresivamente insuficiente para capturar la dimensión situada, emergente y contextual del aprendizaje humano. La literatura crítica en tecnología educativa ha advertido que estos modelos tienden a simplificar en exceso los procesos formativos, generando riesgos de reducción pedagógica y de dependencia excesiva de automatismos [1, 19]. En consecuencia, la falta de integración entre disciplinas deriva en plataformas que no explotan plenamente el potencial adaptativo que los ALS prometen y que la investigación actual demuestra

como alcanzable [20, 21, 22].

Esta fragmentación disciplinar genera desarrollos aislados, inconsistencias conceptuales, vocabularios técnicos y pedagógicos incommensurables, y arquitecturas de software que no responden a las necesidades reales de adaptación pedagógica. Los sistemas resultantes son rígidos, difíciles de mantener, imposibles de generalizar, y pedagógicamente limitados, perpetuando la brecha entre investigación y práctica. Existe, por tanto, una necesidad crítica de **marcos arquitectónicos que operen como artefactos puente**, capaces de traducir principios pedagógicos en decisiones arquitectónicas implementables, facilitar la comunicación entre disciplinas, unificar vocabularios técnicos y educativos, y habilitar transformaciones sistemáticas y reproducibles de LMS hacia ALS, sin requerir reescrituras completas, sin comprometer la continuidad operativa, y garantizando interoperabilidad con estándares educativos internacionales [23, 24].

La presente tesis doctoral aborda esta problemática mediante la construcción, aplicación e implementación de una **Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativos** denominada **ARALS** (*Adaptive Reference Architecture for Learning Systems*), que opera simultáneamente como: (1) *marco conceptual* que formaliza el conocimiento del dominio LMS y ALS, (2) *guía metodológica* que estructura el proceso de transformación arquitectónica, (3) *artefacto de conocimiento* que captura patrones, decisiones y restricciones validadas, y (4) *artefacto puente interdisciplinario* que traduce conceptos pedagógicos en componentes arquitectónicos, facilita la comunicación entre pedagogos, analistas, desarrolladores e implementadores, y evita desarrollos ad-hoc y fragmentados. Esta investigación se posiciona en la intersección de múltiples disciplinas: ingeniería de software, tecnologías adaptativas, tecnología educativa, integraciones afines con el objetivo último de reducir la brecha entre investigación y práctica, y entre disciplinas que deben colaborar para construir sistemas educativos verdaderamente adaptativos.

1.2. Planteamiento del Problema

El problema central que aborda esta tesis puede formularse mediante la siguiente pregunta de investigación:

¿Cómo transformar sistemáticamente un Learning Management System (LMS) tradicional en un Adaptive Learning System (ALS) moderno mediante un mecanismo de representación, a través de transformaciones a nivel arquitectónico, basado en una arquitectura de referencia que facilite

Planteamiento del Problema

el diseño e implementación de funcionalidades y propiedades adaptativas de aprendizaje, y que opere como artefacto puente entre disciplinas del campo de la ingeniería informática y expertos en plataformas educativa, reduciendo la brecha conceptual y metodológica entre pedagogos, analistas funcionales, desarrolladores e implementadores?

Esta pregunta reconoce explícitamente que el desafío de construir ALS no es únicamente técnico, sino fundamentalmente interdisciplinario. Múltiples estudios señalan que la separación entre las necesidades pedagógicas —personalización, adaptación, retroalimentación contextualizada, andamiaje y evaluación formativa— y las soluciones técnicas implementadas —interfaces estandarizadas, flujos transaccionales rígidos, modelos de datos sin semántica educativa y algoritmos de recomendación descontextualizados— constituye un obstáculo estructural. Este problema se evidencia también en investigaciones sobre arquitecturas de sistemas adaptativos, que documentan la dificultad de articular los principios pedagógicos con implementaciones técnicas coherentes y recomiendan marcos de referencia que guíen dicha integración [20, 21, 22]. Revisiones sistemáticas de arquitecturas adaptativas subrayan que las propuestas aisladas o ad hoc tienden a fragmentarse y a perder coherencia, lo que refuerza la necesidad de un enfoque estructurado [17, 18]. Frente a esta situación, la presente tesis sostiene que la arquitectura es un camino verosímil entre otros posibles, porque proporciona un marco de referencia que no solo especifica componentes y patrones de diseño, sino que también traduce principios pedagógicos en restricciones arquitectónicas verificables, alinea vocabularios técnicos y educativos y facilita la colaboración efectiva entre roles con formaciones y objetivos distintos. Trabajos recientes de modelado y diseño de sistemas de aprendizaje adaptativo muestran que esta traducción es indispensable para lograr mejoras significativas [22].

No se descartan otras estrategias complementarias —como la implementación de nuevos servicios integrados por algún componente tecnológico adecuado o la creación de contenidos adaptativos—, pero se argumenta que una **arquitectura de referencia adaptativa** constituye el vector fundamental para materializar funcionalidades y propiedades adaptativas en los LMS, ya que permite traducir principios pedagógicos en decisiones de diseño verificables y orquestar de forma coherente las iniciativas complementarias [25, 22]. La ingeniería de software ha demostrado que abordar los requisitos en el nivel arquitectónico favorece la modularidad, la escalabilidad y la capacidad de análisis de calidad, al incorporar marcos de evaluación como el ATAM y QAW [26, 27], estándares de referencia que definen modelos y métricas de calidad [28, 29], así como criterios de descripción arquitectónica y procesos formales como los propuestos por

ISO/IEC/IEEE 42010 y 42020 [30, 31]. Tales marcos permiten identificar y gestionar atributos críticos (fiabilidad, mantenibilidad, interoperabilidad) desde la concepción del sistema, evitando la dispersión típica de las soluciones ad hoc. En contraste, algunas aproximaciones modernas centradas en la implementación —por ejemplo, las arquitecturas basadas en microservicios que descomponen un LMS en servicios independientes [32, 33]— aportan flexibilidad técnica pero no abordan explícitamente la adaptación pedagógica ni reducen la brecha interdisciplinaria, pudiendo derivar en infraestructuras fragmentadas y difíciles de armonizar con modelos de usuario o ciclos de adaptación como MAPE-K [34, 35]. Por ello, situar el esfuerzo de evolución en la arquitectura de referencia adaptativa ofrece un camino verosímil y sostenible para integrar tecnología y pedagogía de manera transversal.

Esta pregunta central se descompone en las siguientes interrogantes específicas que estructuran la investigación:

P1. ¿Por qué una arquitectura de referencia para LMS? La diversidad de plataformas LMS existentes (Moodle, Open edX, Sakai, Chamilo, Claroline, Canvas, Blackboard, entre otras) presenta arquitecturas heterogéneas, tecnologías diversas y modelos de diseño diferenciados [3, 36]. Esta heterogeneidad dificulta la comparación sistemática de capacidades, la identificación de brechas funcionales, la transferencia de mejoras entre plataformas, y la evaluación de madurez arquitectónica. Más aún, cada plataforma ha sido desarrollada por equipos con distintas formaciones disciplinares, resultando en soluciones que reflejan sesgos conceptuales: algunas priorizan gestión administrativa (influencia de dominios ERP), otras enfatizan colaboración social (influencia de redes sociales), y otras se centran en distribución de contenidos (influencia de sistemas de gestión documental). Una arquitectura de referencia permite *abstraer los elementos comunes y diferenciales* de estas plataformas, caracterizarlas mediante un modelo conceptual unificado, identificar componentes arquitectónicos esenciales, y establecer criterios objetivos de evaluación [9, 10].

Esencialmente, una arquitectura de referencia actúa como *instrumento de conocimiento* que captura patrones arquitectónicos, decisiones de diseño, restricciones de calidad y mejores prácticas del dominio, facilitando la construcción de nuevas arquitecturas concretas o la evolución de arquitecturas existentes [15, 37]. En el contexto de esta investigación, la arquitectura de referencia LMS opera como *base conceptual compartida* que permite a pedagogos, analistas y desarrolladores razonar sobre el mismo dominio con vocabulario consistente, identificar qué elementos son esenciales (núcleo invariante del dominio LMS) y qué elementos son variables (puntos de extensión para adaptación), y aplicar mejoras generalizables a cualquier LMS que la arquitectura

Planteamiento del Problema

represente, sin requerir análisis ad-hoc para cada plataforma. Esto resuelve inconsistencias conceptuales producidas por desarrollos aislados y habilita la construcción sistemática de soluciones adaptativas transferibles.

P2. ¿Cómo articular la transformación LMS→ALS? La transición de un LMS tradicional hacia un ALS no puede concebirse como una migración abrupta o una sustitución tecnológica completa. Las instituciones educativas han invertido recursos significativos en sus plataformas actuales, han desarrollado contenidos digitales, han capacitado a docentes y estudiantes, y han integrado sistemas institucionales (gestión académica, bibliotecas, repositorios) [23, 38]. Una transformación viable debe ser *incremental, evolutiva y preservadora de continuidad operativa*, permitiendo que las funcionalidades críticas del LMS se mantengan operativas mientras se incorporan gradualmente capacidades adaptativas [39, 40].

Esto requiere una *secuencia estratégica de transformación* que reduzca complejidad, habilite trazabilidad, y formalice el proceso de evolución arquitectónica. La secuencia propuesta en esta tesis opera como sigue: (1) se parte de un *LMS concreto* (Moodle, Open edX, Sakai, etc.) con su arquitectura específica; (2) se *abstrae su arquitectura* identificando componentes, capas funcionales, patrones de diseño y estilos arquitectónicos; (3) se *generaliza* mediante una *arquitectura de referencia LMS* que captura elementos comunes de múltiples plataformas, eliminando detalles implementacionales y reteniendo el núcleo conceptual del dominio; (4) se *extiende* la arquitectura de referencia incorporando componentes adaptativos (motores de inferencia, modelos de usuario, bucles MAPE-K, repositorios de conocimiento pedagógico), dando lugar a *ARALS* (Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativos); (5) se *instancia* ARALS en un *ALS concreto* mediante transformaciones arquitectónicas incrementales que preservan servicios críticos y adicionan capacidades adaptativas en fases controladas.

Esta secuencia estratégica garantiza que las mejoras representadas en ARALS (componentes adaptativos, mecanismos de inteligencia, modelos de usuario multidimensionales, ciclos de retroalimentación) puedan aplicarse sistemáticamente a *cualquier* LMS que la arquitectura de referencia represente, sin requerir diseños ad-hoc para cada plataforma, habilitando *generalización* (aplicabilidad amplia), *replicabilidad* (reproducibilidad del proceso), *transferibilidad* (portabilidad de soluciones), y *trazabilidad* (seguimiento de decisiones arquitectónicas desde principios pedagógicos hasta implementaciones concretas). Más aún, esta secuencia facilita la incorporación incremental de propiedades adaptativas, permitiendo que instituciones con distintos niveles de madurez tecnológica y recursos disponibles avancen a su propio ritmo, validando

cada etapa antes de proceder a la siguiente.

P3. ¿Qué propiedades definen un ALS genuino? La literatura sobre sistemas adaptativos presenta definiciones heterogéneas y frecuentemente ambiguas de lo que constituye "adaptación" en contextos educativos [5, 6]. Algunas propuestas reducen la adaptación a recomendaciones de contenidos, otras a ajustes de dificultad en evaluaciones, y otras a personalización de interfaces. Esta ambigüedad conceptual refleja, en parte, la desconexión entre perspectivas pedagógicas (que priorizan adaptación a estilos de aprendizaje, autorregulación, andamiaje) y perspectivas técnicas (que priorizan algoritmos de filtrado colaborativo, sistemas de recomendación, interfaces responsivas).

Esta investigación sostiene que un ALS genuino debe satisfacer un conjunto articulado de **propiedades adaptativas** que abarcan dimensiones técnicas, pedagógicas, cognitivas, sociales y éticas. Estas propiedades no son requisitos funcionales aislados, sino *principios arquitectónicos estructurantes* que definen el comportamiento esperado del sistema, derivan en restricciones de diseño, guían decisiones de implementación, y establecen criterios de validación [15, 16]. Esencialmente, estas propiedades operan como *punto conceptual* entre pedagogos (que especifican necesidades de adaptación) y desarrolladores (que implementan mecanismos técnicos): cada propiedad se define en términos pedagógicos (qué significa adaptar), se traduce en requisitos arquitectónicos (qué componentes y patrones son necesarios), y se operacionaliza en métricas verificables (cómo evaluar si la adaptación es efectiva).

Entre estas propiedades, la **Personalización Adaptativa Colaborativa (PAC)** emerge como propiedad núcleo que sintetiza la noción de adaptación como proceso continuo, dinámico, contextual, colaborativo y pedagógicamente fundamentado. PAC integra tres dimensiones: (a) adaptación como proceso continuo y no lineal (no un ajuste puntual, sino un ciclo de monitoreo-análisis-planificación-ejecución), (b) personalización como equilibrio entre preferencias individuales y objetivos institucionales (no individualismo extremo, sino personalización situada en contextos sociales y curriculares), (c) colaboración como fuente de enriquecimiento contextual, involucrando tanto colaboración humana (docentes, pares, tutores) como colaboración computacional (agentes inteligentes, sistemas de recomendación, modelos de lenguaje). Las demás propiedades adaptativas (interoperabilidad, escalabilidad, retroalimentación, multistakeholder, cognición, accountability, evaluación) se articulan en torno a PAC, operando como restricciones que garantizan que la adaptación sea técnicamente viable, pedagógicamente fundamentada, éticamente responsable, y socialmente situada.

P4. ¿Cómo garantizar calidad, escalabilidad e interoperabilidad? Los sis-

Planteamiento del Problema

temas educativos operan en contextos institucionales complejos, con requisitos estrictos de disponibilidad, seguridad, privacidad, escalabilidad y cumplimiento normativo [41, 42]. Un ALS debe satisfacer atributos de calidad arquitectónica tales como: escalabilidad (capacidad de soportar crecimiento de usuarios y datos sin degradación de desempeño), mantenibilidad (facilidad de evolución, corrección y extensión del sistema), interoperabilidad (integración con sistemas externos mediante estándares abiertos), seguridad (protección de datos sensibles y cumplimiento de regulaciones de privacidad), usabilidad (experiencia de usuario satisfactoria para múltiples perfiles: estudiantes, docentes, administradores), y explicabilidad (transparencia de decisiones algorítmicas, trazabilidad de recomendaciones adaptativas) [15, 28].

Estos atributos deben operacionalizarse mediante *decisiones arquitectónicas explícitas*, documentadas y trazables. Aquí, nuevamente, la arquitectura de referencia ARALS opera como artefacto puente: cada atributo de calidad se vincula con propiedades adaptativas (nivel pedagógico), se traduce en requisitos arquitectónicos (nivel de diseño), se implementa mediante patrones de diseño validados (nivel técnico), y se evalúa mediante mecanismos basados en estándares internacionales (ISO/IEC 25010, ISO/IEC 25023) [43, 44]. Esta trazabilidad multinivel garantiza que las decisiones técnicas (por ejemplo, adoptar arquitectura de microservicios para escalabilidad) estén justificadas por requisitos pedagógicos (por ejemplo, soportar picos de demanda durante períodos de evaluación sin afectar la experiencia de aprendizaje), y que las evaluaciones de calidad consideren tanto métricas técnicas (latencia, throughput, disponibilidad) como métricas pedagógicas (efectividad de adaptación, satisfacción de estudiantes, logro de objetivos de aprendizaje).

P5. ¿Cómo abordar las dimensiones éticas y de gobernanza? La automatización de decisiones pedagógicas mediante algoritmos de inteligencia artificial plantea desafíos éticos significativos: sesgos algorítmicos que reproducen inequidades históricas, erosión de la agencia docente y estudiantil, homogeneización de experiencias educativas, opacidad de criterios de adaptación, y riesgos de vigilancia y explotación de datos personales [1, 19, 45]. Estos desafíos son particularmente críticos en el contexto de la brecha interdisciplinaria: desarrolladores sin formación pedagógica pueden implementar algoritmos técnicamente sofisticados pero pedagógicamente inapropiados, mientras que pedagogos sin comprensión técnica pueden desconocer los sesgos implícitos en los datos de entrenamiento o las limitaciones de los modelos predictivos.

Un ALS responsable debe incorporar mecanismos de *accountability* (rendición de cuentas), trazabilidad de decisiones, explicabilidad de recomendaciones, auditoría algorítmica, y gobernanza participativa que involucre a docentes, estudiantes e institu-

ciones en la definición de políticas adaptativas [42, 46]. ARALS, como artefacto puente, debe facilitar la comunicación sobre dilemas éticos entre roles: pedagogos deben poder expresar principios éticos (equidad, respeto a la diversidad, promoción de autonomía) en términos que analistas puedan traducir en requisitos (auditoría de sesgos, mecanismos de apelación, transparencia de criterios), que desarrolladores puedan implementar en mecanismos técnicos (logging de decisiones, explicabilidad de modelos, interfaces de auditoría), y que implementadores puedan integrar en flujos operativos (procesos de revisión ética, comités de gobernanza, protocolos de respuesta a incidentes).

La respuesta a estas cinco interrogantes requiere un marco conceptual y metodológico que integre: (1) una arquitectura de referencia que abstraiga y estructure el conocimiento del dominio LMS, (2) un proceso sistemático de transformación arquitectónica fundamentado en estándares de ingeniería de software, (3) un conjunto de propiedades adaptativas que operacionalicen la noción de ALS y operen como puente entre pedagogía e ingeniería, (4) mecanismos de validación arquitectónica que garanticen calidad y conformidad, (5) principios de gobernanza ética que aseguren responsabilidad y transparencia, y (6) artefactos interdisciplinarios (modelos, diagramas, matrices de trazabilidad, vocabularios compartidos) que faciliten la colaboración efectiva entre pedagogos, analistas, desarrolladores e implementadores.

1.3. Objetivos de la Investigación

Para dar respuesta a la problemática identificada y abordar la brecha interdisciplinaria en el desarrollo de sistemas de aprendizaje adaptativos, se han definido los siguientes objetivos que guían esta investigación.

1.3.1. Objetivo General

Diseñar, construir, implementar y validar una **Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativos (ARALS)** que permita transformar sistemáticamente plataformas *Learning Management Systems* (LMS) tradicionales en *Adaptive Learning Systems* (ALS), mediante un proceso metodológicamente riguroso, técnicamente escalable, pedagógicamente fundamentado y éticamente responsable, que garantice interoperabilidad con estándares educativos internacionales, preservación de continuidad operativa, validación empírica de atributos de calidad arquitectónica, y que opere como *artefacto puente interdisciplinario* capaz de reducir la brecha conceptual y metodológica entre pedagogos, analistas funcionales, desarrolladores e im-

plementadores, facilitando la traducción de principios pedagógicos en decisiones arquitectónicas implementables y la construcción colaborativa de sistemas adaptativos genuinos.

1.3.2. Objetivos Específicos

Para la consecución del objetivo general anteriormente descrito, se han definido los siguientes objetivos específicos que guían las fases analíticas, de diseño y de validación de la propuesta:

OE1. Caracterizar arquitectónicamente las plataformas LMS existentes mediante el análisis sistemático de sus componentes, capas funcionales, patrones de diseño, estilos arquitectónicos y capacidades adaptativas actuales, identificando elementos comunes (núcleo invariante del dominio), diferenciales (variabilidad entre plataformas) y brechas respecto a las funcionalidades requeridas en un ALS, con énfasis en comprender cómo distintos sesgos disciplinares (gestión administrativa, colaboración social, distribución de contenidos) han influido en las decisiones arquitectónicas de cada plataforma [3, 36, 47].

OE2. Definir y formalizar un conjunto de propiedades adaptativas que operacionalicen la noción de *Adaptive Learning System*, estableciendo para cada propiedad: fundamentación teórica (teorías del aprendizaje, modelos cognitivos), justificación técnica (requisitos arquitectónicos), criterios de validación (métricas de calidad), y vinculación explícita con componentes de ARALS. En particular, formalizar la **Propiedad PAC (Personalización Adaptativa Colaborativa)** como propiedad núcleo que integra adaptación continua, personalización equilibrada y colaboración contextual, y demostrar cómo cada propiedad opera como *punto conceptual* entre necesidades pedagógicas y soluciones técnicas [5, 6, 8].

OE3. Diseñar la arquitectura de referencia denominada en esta tesis como ARALS estructurada en capas funcionales (Presentación, Servicios, Adaptación Inteligente, Dominio/Modelo, Datos, Integraciones Externas), especificando para cada capa: componentes arquitectónicos, responsabilidades funcionales, interfaces de comunicación, patrones de diseño aplicables, y mecanismos de adaptación basados en el ciclo MAPE-K (*Monitor-Analyze-Plan-Execute-Knowledge*), garantizando que ARALS capture tanto el conocimiento técnico

(decisiones arquitectónicas, patrones, restricciones) como el conocimiento pedagógico (principios de adaptación, teorías del aprendizaje, criterios de efectividad) en un marco unificado accesible para múltiples disciplinas [34, 48, 15].

- OE4. Establecer una hoja de ruta metodológica** para la transformación LMS→ALS, estructurada en fases secuenciales con entregables verificables, criterios de entrada/salida, actividades específicas, riesgos identificados y estrategias de mitigación, fundamentada en principios de gestión de proyectos (PMBOK), gestión de procesos de negocio (BPM), y arquitecturas empresariales (TOGAF), explicitando en cada fase los roles de pedagogos, analistas, desarrolladores e implementadores, y proporcionando artefactos de comunicación (matrices de trazabilidad, diagramas de decisiones, vocabularios compartidos) que faciliten la colaboración interdisciplinaria efectiva [49, 50, 51].
- OE5. Validar la arquitectura ARALS** mediante: (a) análisis de cobertura de componentes respecto a plataformas LMS reales, demostrando que ARALS captura los elementos esenciales del dominio; (b) evaluación de conformidad con estilos arquitectónicos y patrones de diseño reconocidos; (c) verificación de atributos de calidad mediante métodos de evaluación arquitectónica (ATAM, SAAM); (d) implementación de prototipos funcionales que demuestren viabilidad técnica de componentes adaptativos clave; (e) estudios de caso en contextos educativos reales que evidencien mejoras en resultados de aprendizaje, y (f) evaluación de la efectividad de ARALS como artefacto puente mediante estudios de usabilidad con pedagogos, analistas y desarrolladores, midiendo comprensibilidad, utilidad percibida, y facilitación de comunicación interdisciplinaria [43, 44, 9].
- OE6. Contribuir al conocimiento científico** en ingeniería de software aplicada a educación mediante: (a) un catálogo de patrones arquitectónicos para sistemas adaptativos educativos, documentados con justificación pedagógica y técnica; (b) una matriz de trazabilidad propiedades-requisitos-métricas que vincule niveles conceptual, arquitectónico y operativo; (c) un marco de validación multinivel (estático, dinámico, empírico) aplicable a arquitecturas de sistemas educativos; (d) lineamientos de gobernanza ética para ALS basados en IA, articulando principios éticos con mecanismos técnicos; (e) recomendaciones para la adopción institucional sostenible de tecnologías adaptativas, considerando factores técnicos, organizacionales, pedagógicos y económicos, y (f) un conjunto de artefactos de comunicación interdisciplinaria (glosarios, matrices de traducción concepto-componente, guías de facilitación) que puedan ser adoptados por

equipos multidisciplinares en proyectos de desarrollo de tecnología educativa.

OE7. Reducir la brecha conceptual y metodológica entre pedagogos, analistas funcionales, desarrolladores e implementadores mediante la construcción y validación de artefactos arquitectónicos compartidos que: (a) traduzcan principios pedagógicos (teorías del aprendizaje, objetivos educativos, criterios de efectividad) en decisiones arquitectónicas verificables (componentes, interfaces, patrones, restricciones); (b) alineen vocabularios técnicos y educativos mediante glosarios formales, ontologías de dominio y matrices de correspondencia concepto-término; (c) faciliten la comunicación efectiva entre roles mediante notaciones visuales accesibles (diagramas arquitectónicos, mapas de decisiones, matrices de trazabilidad) que abstraigan complejidad técnica sin perder precisión; (d) establezcan procesos colaborativos de elicitación, diseño y validación que involucren activamente a representantes de cada disciplina en todas las etapas del ciclo de desarrollo; y (e) proporcionen mecanismos de verificación cruzada que permitan a pedagogos validar que las implementaciones técnicas respetan principios pedagógicos, y a desarrolladores validar que los requisitos pedagógicos son técnicamente viables, promoviendo así la construcción de sistemas adaptativos que sean simultáneamente rigurosos técnicamente y fundamentados pedagógicamente.

1.4. Contribución

La presente investigación realiza contribuciones originales en múltiples dimensiones del conocimiento científico y técnico, con énfasis particular en la facilitación de colaboración interdisciplinaria para el desarrollo de sistemas educativos adaptativos a través de transformaciones de sistemas de gestión de aprendizaje tradicionales:

C1. Secuencia de transformación conceptual LMS→ALS mediante arquitectura de referencia. Esta tesis propone y fundamenta una secuencia conceptual que constituye el núcleo epistemológico de la investigación:

$$LMS \text{ tradicionales} \rightarrow \text{Arquitectura LMS} \rightarrow \text{Arquitectura de Referencia LMS} \rightarrow \text{Arquitectura de Referencia (ARALS)} \rightarrow \text{ALS genuino}$$

Esta secuencia se despliega a través de las siguientes etapas: (1) Se parte de un LMS concreto (Moodle, Open edX, Sakai, etc.) con su arquitectura específica; (2) Se abstrae su arquitectura identificando componentes, capas y patrones; (3) Se generaliza

mediante una arquitectura de referencia que captura elementos comunes de múltiples LMS, eliminando detalles implementacionales y reteniendo el núcleo conceptual del dominio; (4) Se extiende la arquitectura de referencia incorporando componentes adaptativos (motores de inferencia, modelos de usuario, bucles MAPE-K, repositorios de conocimiento pedagógico), dando lugar a ARALS; (5) Se instancia ARALS en un ALS concreto mediante transformaciones arquitectónicas incrementales que preservan servicios críticos y adicionan capacidades adaptativas en fases controladas.

El valor metodológico de esta secuencia se sustenta en que las mejoras representadas en ARALS (componentes adaptativos, mecanismos de inteligencia, modelos de usuario) pueden aplicarse sistemáticamente a *cualquier* LMS que la arquitectura de referencia represente, sin requerir diseños ad-hoc para cada plataforma. Esto garantiza *generalización* (aplicabilidad amplia), *replicabilidad* (reproducibilidad del proceso), *transferibilidad* (portabilidad de soluciones), y *trazabilidad* (seguimiento de decisiones desde principios pedagógicos hasta implementaciones concretas). Más aún, esta secuencia reduce la complejidad cognitiva del proceso de transformación, habilitando que equipos interdisciplinarios razonen sobre transformaciones arquitectónicas en niveles de abstracción apropiados: pedagogos pueden enfocarse en propiedades adaptativas deseadas (nivel conceptual), analistas pueden trabajar en requisitos arquitectónicos (nivel de diseño), y desarrolladores pueden implementar componentes concretos (nivel técnico), manteniendo trazabilidad entre niveles [9, 10].

C2. Propiedad PAC (Personalización Adaptativa Colaborativa) como principio arquitectónico estructurante y puente interdisciplinario. Se introduce y formaliza la Propiedad PAC como propiedad núcleo de los ALS, integrando tres dimensiones: (a) adaptación como proceso continuo, dinámico y no lineal, (b) personalización como equilibrio entre preferencias individuales y objetivos institucionales, (c) colaboración como fuente de enriquecimiento contextual (humano y computacional). PAC actúa como *propiedad pivote* de la cual dependen y hacia la cual convergen las demás propiedades adaptativas (interoperabilidad, escalabilidad, retroalimentación, multistakeholder, cognición, gobernanza, evaluación).

Crucialmente, PAC opera como *artefacto puente* entre pedagogía e ingeniería: se define en términos pedagógicos comprensibles para educadores (adaptación situada, personalización equilibrada, colaboración auténtica), se traduce en requisitos arquitectónicos comprensibles para analistas (componentes de modelado de usuario, motores de inferencia, repositorios de conocimiento), y se implementa mediante patrones de diseño comprensibles para desarrolladores (Observer para monitoreo de eventos, Strategy para políticas adaptativas, Repository para perfiles de usuario). Esta traducción

multinivel garantiza que todos los roles involucrados comprendan qué significa "personalización adaptativa colaborativa" desde sus respectivas perspectivas disciplinares, facilitando comunicación efectiva y validación cruzada [46].

C3. Concepto de Evaluabilidad Adaptativa. Se propone el concepto de *Evaluabilidad Adaptativa* como capacidad del sistema de evaluar continuamente la efectividad pedagógica de sus propias decisiones adaptativas, permitiendo mejora iterativa basada en evidencias. Este concepto trasciende la evaluación adaptativa (ajuste de evaluaciones al progreso del estudiante) para incorporar metaevaluación (evaluación de la calidad de los instrumentos adaptativos mismos). La Evaluabilidad Adaptativa habilita ciclos de mejora continua en los que pedagogos pueden revisar evidencias de efectividad de adaptaciones, analistas pueden identificar patrones de fallas o sesgos, y desarrolladores pueden refinar algoritmos, promoviendo colaboración basada en datos empíricos [52, 53].

C4. Arquitectura de Referencia ARALS como artefacto puente interdisciplinario. Diseño completo de una arquitectura de referencia para ALS, estructurada en seis capas funcionales con 25+ componentes arquitectónicos especificados en términos de responsabilidades, interfaces, tecnologías recomendadas, patrones de diseño aplicables y mecanismos de adaptación. ARALS integra el ciclo MAPE-K como marco conceptual de control autónomo, modelos de usuario multidimensionales (cognitivo, emocional, social, contextual), repositorios de conocimiento pedagógico, y agentes inteligentes basados en modelos de lenguaje de gran escala.

Más allá de su valor técnico, ARALS opera como *artefacto de comunicación interdisciplinaria*: cada componente se documenta con justificación pedagógica (qué necesidad educativa satisface), justificación técnica (qué problema arquitectónico resuelve), y criterios de validación (cómo evaluar su efectividad). Los diagramas arquitectónicos utilizan notaciones híbridas que combinan elementos de UML (comprensibles para desarrolladores), BPMN (comprensibles para analistas de procesos), y mapas conceptuales (comprensibles para pedagogos), facilitando que cada rol pueda interpretar la arquitectura desde su perspectiva disciplinar sin perder coherencia global. ARALS incluye, además, matrices de trazabilidad que vinculan explícitamente propiedades adaptativas (nivel pedagógico) con componentes arquitectónicos (nivel técnico) y patrones de diseño (nivel implementacional), proporcionando navegabilidad bidireccional entre niveles de abstracción [8, 2].

C5. Catálogo de patrones arquitectónicos para ALS con justificación pedagógica y técnica. Identificación, documentación y análisis de 15+ patrones arquitectónicos específicamente relevantes para sistemas adaptativos educativos (MVC

adaptativo, Repository de perfiles, Observer de eventos de aprendizaje, Strategy de políticas adaptativas, Factory de rutas personalizadas, Adapter de integraciones), con análisis de sus implicancias en atributos de calidad (mantenibilidad, escalabilidad, extensibilidad, testabilidad). Cada patrón se documenta siguiendo una plantilla que incluye: (a) problema pedagógico que resuelve, (b) contexto educativo de aplicación, (c) solución arquitectónica, (d) consecuencias técnicas, (e) ejemplos de implementación en plataformas LMS existentes, y (f) guías de adopción para equipos interdisciplinarios. Esta documentación híbrida facilita que pedagogos comprendan el valor educativo de decisiones arquitectónicas, y que desarrolladores comprendan la fundamentación pedagógica de requisitos técnicos [54, 55].

C6. Matriz de trazabilidad propiedades-requisitos-métricas como artefacto de validación cruzada. Marco operativo que vincula propiedades adaptativas (nivel conceptual), requisitos arquitectónicos (nivel de diseño), atributos de calidad ISO/IEC 25010 (nivel de evaluación) y métricas cuantificables ISO/IEC 25023 (nivel operativo), proporcionando trazabilidad bidireccional entre objetivos pedagógicos y decisiones técnicas. Esta matriz opera como *artefacto de validación cruzada*: pedagogos pueden verificar que sus principios pedagógicos se han traducido correctamente en requisitos, analistas pueden verificar que los requisitos son técnicamente viables, desarrolladores pueden verificar que las implementaciones satisfacen requisitos, y evaluadores pueden verificar que las métricas miden efectivamente los atributos deseados. La matriz facilita, además, la identificación de inconsistencias, omisiones o conflictos entre niveles, promoviendo negociación informada entre disciplinas [28, 29].

C7. Hoja de ruta para transformación LMS→ALS con roles y artefactos interdisciplinarios. Procedimiento metodológico estructurado en seis etapas (Elicitación de propiedades, Análisis arquitectónico as-is, Diseño arquitectónico to-be, Implementación por transformaciones, Validación y métricas, Gobernanza y evolución), con especificación de actividades, entregables, criterios de entrada/salida, riesgos y estrategias de mitigación, fundamentado en PMBOK, BPM y TOGAF. Cada etapa especifica explícitamente: (a) roles involucrados (pedagogos, analistas, desarrolladores, implementadores, stakeholders institucionales), (b) artefactos de entrada (producidos por etapas previas o roles específicos), (c) artefactos de salida (consumidos por etapas posteriores o roles específicos), (d) técnicas de facilitación de comunicación interdisciplinaria (workshops, matrices de decisiones, prototipos de baja fidelidad), y (e) criterios de validación cruzada (mecanismos mediante los cuales cada rol puede verificar que los artefactos producidos por otros roles son consistentes con sus propias perspectivas disciplinares). Esta hoja de ruta no solo guía la transformación técnica, sino que tam-

bién estructura la colaboración interdisciplinaria, reduciendo fricción y malentendidos [49, 50, 51].

C8. Estrategia de validación multinivel con participación interdisciplinaria. Metodología de validación que combina: (a) análisis estáticos (revisiones arquitectónicas, análisis de cobertura de componentes, verificación de conformidad con patrones), (b) análisis dinámicos (pruebas funcionales, evaluación de desempeño, pruebas de estrés, A/B testing), (c) evaluación empírica (experimentos controlados con usuarios, estudios de caso institucionales, análisis de efectividad pedagógica), y (d) evaluación de usabilidad interdisciplinaria (estudios con pedagogos, analistas y desarrolladores para medir comprensibilidad de artefactos, utilidad percibida de ARALS como herramienta de comunicación, y facilitación de colaboración efectiva). Esta estrategia multinivel garantiza que ARALS sea validada no solo técnicamente (¿funciona correctamente?) y pedagógicamente (¿mejora el aprendizaje?), sino también interdisciplinariamente (¿facilita la colaboración entre roles?) [43, 44].

1.4.1. Trabajos Presentados Vinculados con la Tesis

La solidez de las contribuciones presentadas en esta tesis doctoral es el resultado de una trayectoria de investigación de largo aliento que se extiende por casi dos décadas de labor científica sostenida. Los pilares que fundamentan la arquitectura ARALS encuentran su génesis en los estudios iniciales sobre los **Dispositivos Hipermediales Dinámicos (DHD)**, donde ya en 2008 se exploraban las primeras aproximaciones al diseño e implementación de entornos educativos flexibles y participativos [56]. Esta línea de pensamiento evolucionó hacia la formalización de arquitecturas para DHD documentadas en foros internacionales de referencia como el CLEI [57], estableciendo un precedente técnico fundamental sobre la adaptabilidad y la hipermedialidad que constituye la base conceptual de la presente investigación.

A partir de estos cimientos, la línea de investigación se expandió hacia la articulación entre modularidad, escalabilidad e interoperabilidad en plataformas educativas contemporáneas. Los principios arquitectónicos que sustentan el desacoplamiento de servicios en ARALS fueron validados en trabajos previos orientados al diseño de aplicaciones e-learning adaptativas basadas en arquitecturas extensibles [58]. Asimismo, la complejidad de integrar capacidades avanzadas de inteligencia artificial preservando la gobernanza y la evolución controlada del software —aspectos centrales de los ALS modernos— ha sido abordada en investigaciones previas sobre servicios de IA generativa y arquitecturas para el sector público [59, 60].

Este recorrido científico permite consolidar un enfoque centrado en la separación de responsabilidades (*Separation of Concerns*) y la definición de puntos de extensión, aspectos críticos para la transformación sistemática de sistemas legados. De manera complementaria, los estudios interdisciplinarios sobre la intersección entre educación superior, plataformas digitales e IA [61] aportaron la evidencia empírica necesaria sobre las limitaciones de los entornos actuales y la urgencia de marcos integradores como el propuesto en esta tesis.

Finalmente, este proceso de validación acumulativa converge en la arquitectura ARALS, la cual ha sido sometida al escrutinio de la comunidad internacional de ingeniería de software y aceptada para su publicación en la XXIX Ibero-American Conference on Software Engineering (CIbSE 2026) [62]. Dicha aceptación ratifica la relevancia y solidez metodológica de la propuesta frente a los desafíos globales de la personalización educativa a gran escala.

1.5. Estructura de la Tesis

La presente tesis se organiza en siete capítulos que articulan la fundamentación, diseño, implementación y validación de la propuesta, permitiendo niveles de lectura adaptados a audiencias pedagógicas o técnicas:

Capítulo 1: Introducción. Presenta el contexto, la motivación y la brecha interdisciplinaria como obstáculo para el desarrollo de ALS. Define el problema, objetivos y contribuciones, posicionando a ARALS como un artefacto puente para traducir principios pedagógicos en decisiones técnicas.

Capítulo 2: Estado del Arte y Fundamentación Teórica. Revisa críticamente cuatro áreas: (a) evolución y limitaciones de los LMS; (b) teorías del aprendizaje y taxonomías de ALS; (c) metodologías de construcción y evaluación de arquitecturas de referencia; y (d) ética y gobernanza de la IA en educación. Concluye identificando las brechas que justifican la necesidad de marcos arquitectónicos integradores.

Capítulo 3: Fundamentos Conceptuales y Metodológicos. Describe la estrategia de construcción de ARALS mediante un análisis comparativo de cinco LMS representativos. Detalla la transición desde la caracterización de invariantes hasta la formalización del modelo estratificado AR-LMS, su extensión adaptativa mediante el ciclo MAPE-K y la definición de las propiedades PALS.

Capítulo 4: Diseño de ARALS. Constituye el núcleo técnico de la tesis. Define la arquitectura en seis capas interrelacionadas (Presentación, Aplicación, Adaptación Inteligente, Dominio, Datos e Integración) que operacionalizan el ciclo MAPE-K. El

capítulo justifica la selección de patrones (Microkernel, SOA, Capas) y valida conceptualmente la propuesta frente a atributos de calidad como escalabilidad, interoperabilidad y seguridad.

Capítulo 5: Hoja de Ruta para la Transformación. Desarrolla una guía operativa inspirada en PMBOK y estándares de ingeniería de software para transitar de un LMS a un ALS. Estructura el proceso en seis etapas (elicitación, análisis, diseño, implementación, validación y gobernanza), asegurando la trazabilidad entre las propiedades adaptativas (especialmente PAC) y los componentes de ARALS.

Capítulo 6: Caso de Estudio: Odoo eLearning. Documenta la aplicación práctica de la hoja de ruta sobre el módulo *website_slides* de Odoo. Valida empíricamente la arquitectura mediante la refactorización del sistema, la implementación de servicios de modelado del aprendiz y motores de recomendación, y el análisis de alcance de la solución en un entorno productivo.

Capítulo 7: Conclusiones y Trabajo Futuro. Ofrece una discusión crítica de los resultados frente a la hipótesis inicial. Sistematiza las contribuciones teóricas, arquitectónicas, metodológicas y empíricas, reconoce las limitaciones del estudio y proyecta líneas de investigación hacia la IA generativa y la validación multiplataforma.

Capítulo 2

Estado del Arte y Fundamentación Teórica

2.1. Introducción

En el actual ecosistema de la educación digital, la necesidad de infraestructuras tecnológicas capaces de ajustarse dinámicamente a la singularidad cognitiva y procedimental del estudiante ha adquirido una relevancia estratégica sin precedentes. Este capítulo tiene como propósito fundamental establecer el andamiaje teórico-científico y sistematizar el estado del conocimiento respecto a las arquitecturas de software para sistemas de aprendizaje adaptativos (*Adaptive Learning Systems*, ALS). A diferencia de los entornos tradicionales, donde predomina una visión estática del proceso formativo, la construcción de ALS requiere una convergencia disciplinar que integre la robustez de la ingeniería de software con la flexibilidad de los modelos pedagógicos contemporáneos.

La estructura analítica de este capítulo se articula en tres dimensiones interconectadas que garantizan una comprensión profunda del dominio. En primer lugar, se desarrolla un marco teórico integral que vincula los principios fundamentales de la arquitectura de software —tales como la mantenibilidad y la escalabilidad— con teorías de adaptación computacional de alta complejidad. Posteriormente, se presenta una revisión sistemática de la literatura, basada en el protocolo PRISMA, para caracterizar el panorama tecnológico vigente. Esta revisión es crítica, dado que la evidencia empírica sugiere que, aunque los *Learning Management Systems* (LMS) convenciona-

les dominan el mercado institucional, menos del 15 % de estas plataformas incorporan de manera nativa mecanismos de adaptación inteligente, limitándose en su mayoría a una gestión transaccional de contenidos [3, 4, 5].

Finalmente, el capítulo culmina con un examen riguroso de trabajos relacionados, estableciendo un análisis comparativo que permite detectar las brechas de diseño en las soluciones actuales. Este abordaje estructurado no solo proporciona una base de conocimiento sólida, sino que justifica la necesidad de la arquitectura ARALS como una respuesta a la fragmentación observada en el estado del arte. Al fundamentar la propuesta tanto en principios de diseño de software como en evidencia estadística actualizada sobre el rendimiento y limitaciones de los sistemas existentes [6, 9], se asegura que la arquitectura de referencia resultante sea no solo teóricamente válida, sino técnicamente viable para los desafíos de la educación superior y corporativa actual.

2.2. Marco Teórico

El desarrollo de la arquitectura ARALS requiere una fundamentación multidisciplinar que articule el rigor técnico de la ingeniería de software con la flexibilidad intrínseca de los procesos pedagógicos adaptativos. En esta sección se exponen los pilares conceptuales que sustentan la investigación, organizados en torno a la robustez estructural, las bases del aprendizaje inteligente y los mecanismos de auto-adaptación computacional. Esta base teórica permite situar a ARALS no solo como una propuesta técnica, sino como un artefacto puente capaz de transformar la mediación tecnológica educativa mediante principios de diseño resilientes, evolutivos y pedagógicamente fundamentados.

2.2.1. Fundamentos de Arquitectura de Software

La arquitectura de software se define como la disciplina central de la ingeniería de sistemas encargada de establecer la estructura organizacional y los principios de diseño que rigen la construcción de ecosistemas computacionales complejos. De acuerdo con [15], una arquitectura comprende un conjunto de estructuras que describen los elementos de software, sus propiedades externamente visibles y las relaciones que los vinculan, constituyéndose en un marco de decisiones estratégicas que condiciona la mantenibilidad, escalabilidad y capacidad evolutiva del sistema. En este ámbito, los principios de diseño orientan la creación de soluciones robustas, destacando la *Separación de Responsabilidades* (*Separation of Concerns*) para promover componentes

con funciones delimitadas y acoplamiento mínimo [63]. Este enfoque se complementa con los principios de la *arquitectura limpia* [64], tales como la *Inversión de Dependencias* y el principio *Abierto/Cerrado* [65, 66], los cuales permiten desacoplar niveles de abstracción y diseñar módulos extensibles sin alterar sus implementaciones núcleo, garantizando la independencia frente a la evolución tecnológica.

El valor de una arquitectura se manifiesta a través de sus atributos de calidad o propiedades no funcionales. La *escalabilidad* representa la aptitud para sostener incrementos en la demanda sin degradar el rendimiento [67], mientras que la *mantenibilidad* asegura la facilidad para corregir defectos o incorporar funcionalidades [68]. Otros atributos críticos incluyen la *flexibilidad* adaptativa [6], la *interoperabilidad* para la interacción con sistemas externos [69] y la *reusabilidad* de componentes [70]. Para materializar estas propiedades, la práctica profesional recurre a patrones arquitectónicos como el de *Capas* (*Layered Architecture*), que organiza niveles de abstracción jerárquicos [71], o el modelo *Modelo–Vista–Controlador* (MVC) para la separación de la lógica de negocio y la presentación [72]. En arquitecturas modernas, el paradigma de *Microservicios* [73] habilita sistemas desacoplados y escalables de manera autónoma. Estos fundamentos permiten evaluar la propuesta bajo estándares internacionales como *ISO/IEC/IEEE 42010:2022* [41] y métodos como el ATAM [74], fundamentales para asegurar la consistencia y calidad arquitectónica.

2.2.2. Bases Conceptuales de Sistemas de Aprendizaje Adaptativo

Los sistemas de aprendizaje adaptativo (ALS) constituyen la evolución de los entornos educativos tradicionales hacia plataformas capaces de modificar su comportamiento, contenido e interfaz en respuesta a las necesidades y progresos del estudiante [7]. Sus propiedades núcleo incluyen la personalización cognitiva, la adaptabilidad dinámica en tiempo real y la recomendación inteligente de trayectorias formativas. Estos sistemas se fundamentan en teorías del aprendizaje que guían sus mecanismos de ajuste: la teoría del aprendizaje *Mastery* [75], que ajusta el ritmo de instrucción para asegurar el dominio; la teoría de la carga cognitiva [13], que optimiza la presentación de información para evitar la saturación; y el aprendizaje autorregulado [76], que apoya procesos metacognitivos. Asimismo, el concepto de *zona de desarrollo próximo* [11] es esencial para calibrar el nivel de dificultad y el andamiaje (*scaffolding*) necesario en cada intervención.

A pesar de la solidez teórica de estos modelos, el análisis de las plataformas actuales

revela una brecha tecnológica significativa. Mientras que el 95 % de las instituciones de educación superior dependen de un *Learning Management System* (LMS), la evidencia estadística sugiere que menos del 15 % de estas soluciones comerciales incorporan mecanismos nativos de inteligencia adaptativa, limitándose mayoritariamente a una gestión transaccional y administrativa de contenidos [9, 10]. El núcleo funcional de un ALS genuino requiere modelos computacionales del estudiante que representen formalmente el estado de conocimiento, como el marco cognitivo *ACT-R* [77], los *espacios de conocimiento* [78] para la navegación adaptativa basada en prerrequisitos, y el *knowledge tracing* [79] para la estimación probabilística del dominio de competencias. La integración de estos fundamentos define el marco sobre el cual ARALS busca trascender la rigidez estructural de los sistemas legados.

2.2.3. Teorías de Adaptación en Sistemas Computacionales

La adaptación computacional se apoya en la teoría de sistemas adaptativos complejos, la teoría de control y la ingeniería de software auto-adaptativa. Los sistemas complejos proporcionan el marco para comprender cómo emergen comportamientos globales a partir de interacciones locales, regidos por principios de autoorganización, retroalimentación y no-linealidad [80, 81, 82]. En este contexto, la emergencia de respuestas adaptativas ante la interacción del estudiante con el entorno tecnológico requiere un enfoque no determinista [83]. Por su parte, la teoría de control adaptativo aporta el rigor formal para ajustar parámetros operativos, utilizando el control predictivo basado en modelos [84] para anticipar el desempeño y el control robusto para garantizar la estabilidad ante perturbaciones en entornos educativos variables [85, 86].

Estas teorías se materializan en arquitecturas auto-adaptativas mediante el bucle de control **MAPE-K** (*Monitor-Analyze-Plan-Execute over Knowledge*), que formaliza el ciclo continuo de detección de cambios y aplicación de ajustes [87]. A este paradigma se suma la *computación autónoma* [34], que persigue sistemas capaces de autoconfigurarse y autooptimizarse. El diseño de tales arquitecturas exige principios de modularidad extrema y desacoplamiento mediante interfaces bien definidas [63] e inyección de dependencias [71]. Es crucial la separación de aspectos, específicamente entre la lógica funcional y la lógica de adaptación [88], permitiendo modificaciones en las políticas adaptativas mediante configuración externa sin recompilación. Finalmente, la adaptatividad depende de la observabilidad lograda a través de instrumentación comprehensiva [89] y métricas que evalúen la efectividad de los ajustes realizados [90].

2.2.4. Principios de Diseño Arquitectónico para la Adaptabilidad

El diseño de arquitecturas orientadas a la adaptabilidad requiere la aplicación de principios específicos que faciliten la modificación dinámica del comportamiento del sistema sin comprometer su integridad estructural. Entre estos principios, la modularidad y el desacoplamiento son esenciales, ya que la descomposición del sistema en componentes independientes favorece la reconfiguración, sustitución o incorporación de nuevas funcionalidades con un impacto mínimo sobre el resto de la arquitectura [70]. La existencia de interfaces bien definidas [63] garantiza que las modificaciones internas de un componente no afecten las interacciones con otros, mientras que la inyección de dependencias [71] permite configurar dinámicamente las relaciones entre módulos, aumentando la flexibilidad y extensibilidad del sistema.

Otro principio clave es la separación de aspectos, que promueve la independencia entre la lógica funcional y la lógica de adaptación. De acuerdo con [88], la separación de la lógica de adaptación permite realizar modificaciones en las políticas adaptativas sin alterar las funcionalidades principales, mientras que la configuración externa de los parámetros [71] posibilita ajustar el comportamiento del sistema sin necesidad de recompilación, lo cual incrementa su capacidad de respuesta ante cambios contextuales. Finalmente, la observabilidad y el monitoreo constituyen pilares fundamentales de la adaptabilidad, ya que un sistema sólo puede ajustarse si es capaz de percibir y evaluar su propio estado. Para ello, es necesario incorporar mecanismos de instrumentación que proporcionen visibilidad sobre el comportamiento interno [89] y definir métricas que permitan evaluar la efectividad de las adaptaciones implementadas [90]. Estos principios, aplicados de manera integrada, permiten construir arquitecturas resilientes, auto-observables y evolutivas, capaces de responder proactivamente a las variaciones del entorno operativo.

2.2.5. Marcos de Referencia Conceptuales

Los marcos de referencia conceptuales estructuran el diseño de los ALS desde una perspectiva metodológica. El modelo AHAM (*Adaptive Hypermedia Application Model*) [91] es uno de los más influyentes al distinguir entre el modelo de dominio (contenido), el modelo de usuario (perfil del aprendiz) y el modelo de adaptación (reglas de ajuste). Esta estructura se complementa con la organización funcional de los Sistemas Tutoriales Inteligentes [92], que divide el sistema en módulos de dominio, estudiante, pedagógico e interfaz, sentando las bases para la separación entre conocimiento e

instrucción. Por último, el estándar *IEEE Learning Technology Systems Architecture* (LTSA) [93] ofrece una arquitectura de referencia estandarizada que garantiza la interoperabilidad y consistencia en los flujos de información, proporcionando las directrices necesarias para que la propuesta ARALS se integre de manera coherente y validable en contextos educativos contemporáneos.

2.3. Estado del Arte: Análisis de la Literatura

La construcción de una arquitectura de referencia para sistemas de aprendizaje adaptativos requiere una fundamentación empírica sólida acerca del estado actual del conocimiento en ingeniería de software aplicada al e-learning, contemplando patrones de adaptación, técnicas de personalización y atributos de calidad estructural. El presente análisis sistematizado de la literatura tiene como propósito identificar, examinar y sintetizar las propuestas arquitectónicas reportadas para entornos educativos, persiguiendo los objetivos de: (1) caracterizar las estructuras de software dominantes en los ALS contemporáneos, (2) relevar los patrones de diseño y las tecnologías empleadas para la inyección de capacidades adaptativas, (3) evaluar la respuesta de dichas arquitecturas ante factores críticos como la escalabilidad, mantenibilidad e interoperabilidad, (4) diagnosticar las limitaciones y brechas técnicas presentes en las soluciones actuales, y (5) consolidar las oportunidades de mejora que sustentan el diseño de ARALS.

Este estudio sistematizado funciona como el cimiento metodológico necesario para validar que ARALS capture el conocimiento técnico del dominio, incorpore las mejores prácticas vigentes y resuelva los desafíos estructurales que no han sido abordados por propuestas previas. Más aún, este análisis permite contrastar la propuesta con las arquitecturas existentes, evidenciando que ARALS representa una síntesis fundamentada y una evolución sistemática del conocimiento acumulado en el campo de la tecnología educativa. Con el fin de garantizar la solidez científica y la trazabilidad del proceso, la selección y el tratamiento de las fuentes documentales se ha alineado con las directrices generales del protocolo PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) [94], proporcionando un marco de trabajo transparente y reproducible para la presente investigación.

2.3.1. Metodología de Selección y Análisis de Fuentes

El procedimiento de análisis se diseñó con el fin de identificar y caracterizar las arquitecturas de software vigentes en sistemas de e-learning adaptativos, examinando específicamente sus patrones, mecanismos de ajuste y atributos de calidad estructural. Siguiendo de manera rigurosa criterios técnicos inspirados en las directrices del protocolo PRISMA, la ejecución del estudio se estructuró en cinco fases correlativas: (1) definición de los ejes de análisis, (2) estrategia de búsqueda de fuentes, (3) establecimiento de criterios de selección, (4) extracción de datos técnicos y (5) síntesis de los hallazgos. La justificación de este análisis bibliográfico para el desarrollo de la arquitectura ARALS reside en tres pilares fundamentales: *primero*, mapear los componentes y patrones recurrentes para construir una arquitectura de referencia con base teórica sólida; *segundo*, identificar las brechas críticas en escalabilidad, interoperabilidad y ética que fundamenten una propuesta de arquitectura extendida; y *tercero*, sustentar empíricamente la efectividad de los mecanismos que se proponen en esta investigación. Complementariamente, se adopta una perspectiva de *arquitecturas evolutivas* [39], priorizando estrategias de transformación para sistemas legados (*legacy*) y su migración incremental hacia modelos adaptativos modernos.

Ejes de Análisis y Dimensión del Estudio

Las dimensiones que guían el tratamiento de la información cubren los aspectos estructurales, tecnológicos y cualitativos que se vinculan directamente con los objetivos estratégicos de ARALS:

- EA1. ¿Cuáles son las configuraciones arquitectónicas predominantes y sus componentes clave?** *Justificación y Vínculo:* Permite identificar el núcleo conceptual común (gestión de contenidos, modelos de usuario, motores de adaptación) y validar la organización por capas de ARALS frente a estilos modernos como microservicios o sistemas dirigidos por eventos.
- EA2. ¿Qué patrones y tecnologías permiten operacionalizar las capacidades de adaptación?** *Justificación y Vínculo:* Facilita el relevamiento de patrones de diseño específicos (como MVC adaptativo, *Observer* o *Strategy*) y estándares de comunicación (LTI, xAPI) que sirven de base para el catálogo de patrones y la arquitectura de integración de la presente tesis.
- EA3. ¿Qué brechas estructurales persisten en las propuestas documentadas actualmente?** *Justificación y Vínculo:* Busca detectar limitaciones en

escalabilidad, explicabilidad algorítmica y gobernanza ética de los datos. Estos hallazgos fundamentan las propiedades clave de ARALS, tales como la propiedad PAC y los mecanismos de rendición de cuentas.

EA4. ¿Qué tendencias tecnológicas emergentes pueden potenciar la sostenibilidad de los ALS? *Justificación y Vínculo:* Explora el impacto de paradigmas como *edge computing*, IA explicable (XAI) y patrones de transformación gradual (*strangler pattern*) para orientar las decisiones de diseño avanzado y las proyecciones de evolución futura de la arquitectura propuesta.

Estrategia de Localización de Fuentes

Para asegurar que la base documental de esta investigación sea actual y pertinente, se definió un período de búsqueda comprendido entre los años 2018 y 2024. Este intervalo resulta estratégico para capturar hitos tecnológicos fundamentales como la consolidación de las arquitecturas de microservicios, el auge de la inteligencia artificial aplicada a la educación, la maduración de los estándares de interoperabilidad (específicamente LTI 1.3 y xAPI) y la aceleración de la transformación digital ocurrida en el escenario post-pandemia. Este marco temporal garantiza que ARALS se sustente en el estado del arte vigente, evitando el sesgo de paradigmas tecnológicos hoy obsoletos. La recolección de información se llevó a cabo en los repositorios académicos de mayor impacto en el área de ingeniería y tecnología educativa: **IEEE Xplore**, **ACM Digital Library**, **Springer Link**, **ScienceDirect** y **Web of Science**.

El proceso de búsqueda se articuló en torno a tres dimensiones conceptuales integradas: (1) sistemas de e-learning adaptativos, (2) arquitectura de software (incluyendo el ciclo MAPE-K y principios de autoadaptación) y (3) atributos de calidad sistémica. La cadena de búsqueda final se construyó mediante la combinación lógica de términos representativos de estas dimensiones, siendo adaptada sintácticamente según los requerimientos de cada base de datos:

```
1 ("adaptive e-learning" OR "adaptive learning system" OR "adaptive educational system"
2   OR "personalized learning" OR "intelligent tutoring system" OR "adaptive
3   hypermedia")
4
5 AND
6
7 ("software architecture" OR "system architecture" OR "reference architecture" OR "
8   architectural pattern" OR "architectural style" OR "architectural design" OR "
9   architectural component" OR "MAPE-K" OR "self-adaptive system")
```

7	AND
8	
9	("scalability" OR "flexibility" OR "adaptability" OR "interoperability" OR "maintainability" OR "quality attribute" OR "non-functional requirement" OR "architectural evolution" OR "microservices" OR "service-oriented architecture")

La estructura de esta cadena se fundamenta en la taxonomía conceptual de ARALS, permitiendo identificar investigaciones que integren simultáneamente la dimensión arquitectónica con la capacidad de adaptación en contextos pedagógicos. El primer grupo de términos delimita el dominio de aplicación, abarcando desde sistemas tutores inteligentes hasta hipermedia adaptativa. El segundo grupo garantiza un enfoque estrictamente arquitectónico y la incorporación de modelos de control autónomo como MAPE-K, permitiendo diferenciar este estudio de investigaciones puramente algorítmicas. Finalmente, el tercer grupo filtra las fuentes en función de los atributos de calidad (como escalabilidad e interoperabilidad) que están alineados con las propiedades de ARALS y los estándares de la norma ISO/IEC 25010. Para lograr una cobertura exhaustiva, esta estrategia se complementó con una búsqueda manual dirigida en las actas de conferencias de alta relevancia (como ICAIS, AIED, LAK e ICALT) y la aplicación de la técnica de *snowballing* para identificar referencias clave a partir de las fuentes primarias seleccionadas.

Selección y Criterios de Calidad de las Fuentes

Para garantizar la relevancia técnica y la calidad metodológica necesarias que fundamenten el desarrollo de ARALS, se definieron criterios específicos de selección y evaluación de las fuentes. Los criterios de inclusión (CI) se establecieron atendiendo a trabajos publicados en el intervalo 2018–2024, abarcando artículos arbitrados en revistas de alto impacto, actas de conferencias internacionales y tesis doctorales redactadas en idioma inglés o español. Temáticamente, se priorizaron aquellos estudios centrados en el e-learning adaptativo o sistemas tutores inteligentes que presentaran una descripción explícita de sus estructuras, diagramas de componentes o decisiones de diseño arquitectónico. Asimismo, se incorporaron investigaciones que proporcionaran detalles precisos sobre los mecanismos de adaptación (a nivel de rutas de aprendizaje, contenidos o interfaces) y que contemplaran la evaluación de atributos de calidad bajo los estándares de la norma ISO/IEC 25010. De manera complementaria, los criterios de exclusión (CE) permitieron descartar estudios de naturaleza puramente teórica que carecieran de una arquitectura implementable o de detalles técnicos y algorítmicos suficientes para su análisis. También se omitieron revisiones secundarias, trabajos con

baja rigurosidad metodológica, publicaciones de extensión mínima o registros duplicados, así como investigaciones pertenecientes a dominios ajenos al ámbito educativo. Finalmente, se excluyeron aquellas propuestas con un enfoque exclusivo en dimensiones pedagógicas sin un componente de software sustancial, al igual que estudios basados en infraestructuras tecnológicas obsoletas o pertenecientes al paradigma previo a la consolidación de la computación en la nube. El proceso de selección de la literatura se organizó siguiendo un flujo de trabajo inspirado en el protocolo PRISMA, estructurado en cuatro fases clave de filtrado y validación. En la fase inicial de identificación, se localizaron 1,847 registros en las bases de datos mencionadas y 63 registros mediante búsqueda manual, resultando en 1,342 registros únicos tras la eliminación de duplicados. Durante la etapa de *screening*, se evaluaron títulos y resúmenes aplicando los criterios de inclusión y exclusión, lo que derivó en el descarte de 1,089 registros por falta de enfoque arquitectónico o inadecuación del dominio, logrando un acuerdo de confiabilidad entre revisores representado por un coeficiente $\kappa = 0.89$. En la fase de elegibilidad, se analizaron 253 textos completos, de los cuales se descartaron 147 por insuficiente detalle técnico o falta de validación empírica. El proceso culminó con la conformación de un corpus final de 106 estudios, los cuales fueron sometidos a una extracción sistemática de datos bibliográficos, tecnológicos y arquitectónicos para sustentar la presente investigación.

2.3.2. Análisis de Arquitecturas Actuales

El panorama de los sistemas de e-learning adaptativos refleja una convergencia entre el diseño estructurado tradicional y los paradigmas distribuidos, priorizando la flexibilidad y la escalabilidad. A continuación se sintetizan las aproximaciones más representativas identificadas en la literatura.

Las arquitecturas en capas destacan por su organización jerárquica y separación de funciones. [20] propone un modelo de cuatro estratos: presentación (interfaces adaptativas), lógica de negocio (motor de recomendación), acceso y persistencia de datos (modelos de estudiante y contenido). Si bien este enfoque garantiza mantenibilidad y orden estructural, su naturaleza jerárquica limita la flexibilidad para integrar estrategias pedagógicas dinámicas en tiempo real.

Por su parte, las arquitecturas orientadas a servicios (SOA) maximizan la interoperabilidad mediante la descomposición funcional. [95] describe un modelo consciente del contexto que desacopla servicios de gestión de contexto, adaptación, contenido y evaluación. Aunque esta aproximación mejora la reutilización, la latencia de red y la

complejidad de orquestación plantean desafíos en escenarios con grandes volúmenes de datos o requisitos de respuesta inmediata.

Los microservicios representan la evolución hacia la autonomía operativa y la granularidad. [96] plantea una estructura modular con servicios especializados (autenticación, analítica y recomendación) coordinados mediante un *API Gateway*. Asimismo, [97] documenta la migración de arquitecturas monolíticas hacia este paradigma, reportando mejoras sustanciales en escalabilidad horizontal y despliegue continuo, a pesar del incremento en la complejidad del monitoreo inter-servicios.

Finalmente, el enfoque *cloud-native* consolida la elasticidad sistémica. [98] propone el uso de contenedorización (Docker), orquestación (Kubernetes) y servicios gestionados para optimizar el rendimiento mediante auto-escalado dinámico. No obstante, la dependencia de infraestructuras externas introduce consideraciones críticas sobre la seguridad y la soberanía de los datos educativos a largo plazo.

2.3.3. Tecnologías y Patrones Emergentes

La inteligencia artificial (IA) y el aprendizaje automático constituyen los pilares de la personalización sistémica, permitiendo inferir patrones de comportamiento y predecir el desempeño educativo. [99] propone el uso de redes neuronales profundas y procesamiento de lenguaje natural (NLP) junto con algoritmos de *clustering* para segmentar estudiantes y optimizar recomendaciones. En este ámbito, destacan tres tendencias críticas identificadas por [100]: la *IA explicable* (XAI) para garantizar la trazabilidad de decisiones [101], el *aprendizaje federado* orientado a la privacidad de datos distribuidos [102] y los motores de personalización híbridos [103].

Complementariamente, las arquitecturas dirigidas por eventos (EDA) facilitan la reactividad inmediata del sistema. Mediante el uso de *event sourcing* y la segregación de responsabilidades (CQRS), es posible capturar cada interacción como un evento inmutable, optimizando el acceso concurrente y permitiendo la adaptación del contenido en tiempo real [104]. Esta capacidad de respuesta se sustenta en estándares de interoperabilidad que aseguran la cohesión en ecosistemas heterogéneos. Modelos como SCORM [105], xAPI [106] y LTI [23] resultan esenciales para el intercambio de recursos y la trazabilidad de los procesos formativos, consolidando infraestructuras educativas abiertas, reutilizables y distribuidas.

2.3.4. Casos de Estudio Representativos

El estudio comparado de arquitecturas implementadas en sistemas reales permite evaluar la aplicabilidad y efectividad de los enfoques teóricos. Entre los casos más representativos se encuentra el sistema MeL [107], que adopta una arquitectura de cuatro capas compuesta por una capa de interfaz con soporte multi-dispositivo, una capa de servicios encargada de la lógica adaptativa, una capa de integración con sistemas externos y una capa de datos para la persistencia de información. Los resultados reportan mejoras del 35 % en el nivel de compromiso estudiantil y un aumento del 28 % en la tasa de completitud de los cursos, aunque el acoplamiento entre capas limita la flexibilidad para modificar componentes de forma independiente.

Otro ejemplo relevante es el sistema *Arigatō* [108], que integra mecanismos de realidad aumentada para el aprendizaje adaptativo. Su arquitectura híbrida combina procesamiento local y en la nube, incorporando adaptación contextual basada en la ubicación y el dispositivo, junto con análisis en tiempo real de las interacciones. Los resultados muestran un incremento del 42 % en la retención de conocimientos y una mejora sustancial en la experiencia de usuario; sin embargo, la dependencia de hardware especializado restringe su escalabilidad y portabilidad.

Finalmente, la plataforma de aprendizaje flexible descrita por [109] representa una aproximación pragmática basada en la extensión del LMS Moodle mediante un conjunto de plugins personalizados. Su arquitectura incorpora una API RESTful [110] para integrar servicios externos y un módulo de analítica de aprendizaje unificado [111], logrando una reducción del 50 % en el tiempo de desarrollo de nuevas funcionalidades. No obstante, la dependencia del núcleo de Moodle impone restricciones sobre las posibilidades de optimización arquitectónica y adaptación dinámica.

2.3.5. Síntesis de Hallazgos y Brechas Identificadas

El análisis sistemático de la literatura y de los casos de estudio revisados permite identificar un conjunto de tendencias consolidadas y brechas persistentes en el campo de las arquitecturas para sistemas de aprendizaje adaptativo.

Entre los hallazgos principales, se destaca una convergencia hacia arquitecturas modulares bajo estilos LAYERS o Service-Oriented (SOA), lo que favorece la mantenibilidad y la escalabilidad. Se observa que el patrón MICROKERNEL ha surgido como una solución recurrente para dotar de extensibilidad a plataformas como Moodle u Odoo, permitiendo la adición de complementos aislados. Asimismo, la adopción de infraestructuras cloud-native y arquitecturas de microservicios ha permitido el aprove-

chamamiento de servicios gestionados y el despliegue continuo. En términos de datos, la integración de estándares de interoperabilidad semántica como xAPI y el uso de Learning Record Stores (LRS) se consolidan como el sustrato tecnológico para capturar evidencias de aprendizaje más allá del paradigma tradicional de contenidos estáticos.

Pese a estos avances, se evidencian brechas significativas que justifican la necesidad de una arquitectura de referencia unificada y evolutiva:

- **Rigidez Arquitectónica (High CRI):** La mayoría de las arquitecturas actuales operan bajo un paradigma de "talla única" (one-size-fits-all), donde la lógica pedagógica está fuertemente acoplada al núcleo del sistema (LMS Core). Esto genera una rigidez estructural que impide la reconfiguración dinámica sin rediseños profundos.
- **Limitaciones en la Adaptación Dinámica:** Persiste una brecha en la ejecución de la adaptabilidad en tiempo de ejecución. Los sistemas actuales carecen de un ciclo de control cerrado (como el ciclo MAPE-K), lo que obliga a realizar reconfiguraciones manuales o reinicios para alterar las rutas de aprendizaje.
- **Carencia de Mecanismos de Evolución No Intrusiva:** No se identifican marcos metodológicos que guíen la transformación de sistemas legados (Legacy Monoliths) hacia sistemas adaptativos sin requerir una reescritura total del código. Existe una ausencia del uso de patrones de transformación como el Strangler Fig Pattern adaptado al dominio educativo.
- **Déficit en Propiedades PALS-A:** Se observa una atención insuficiente a atributos arquitectónicamente influenciados como la Observabilidad Granular, la Cohesión Pedagógica y la Gobernanza Adaptativa (Accountability). Las soluciones suelen enfocarse en la funcionalidad del algoritmo de recomendación, descuidando la infraestructura necesaria para garantizar la trazabilidad y la transparencia de las decisiones algorítmicas.

2.4. Trabajos Relacionados

Esta sección analiza las investigaciones que fundamentan el desarrollo de sistemas de aprendizaje adaptativos (ALS), organizándolas según su abordaje del problema central, sus metodologías y las brechas identificadas que justifican la propuesta de ARALS como artefacto puente.

2.4.1. El Problema Central y su Evolución en el Estado del Arte

El problema central que aborda esta tesis —la transformación sistemática y escalable de un LMS tradicional en un ALS moderno— ha sido identificado de manera recurrente en la literatura especializada. Autores seminales como [7] y [6] fueron de los primeros en señalar la rigidez de los entornos educativos y la necesidad de modelos computacionales que soporten la adaptación dinámica. Esta preocupación evolucionó desde prototipos aislados hacia la búsqueda de arquitecturas sistemáticas. Trabajos como los de [9] y [10] identificaron que la falta de arquitecturas de referencia (RA) consolidadas es un obstáculo crítico para la transferencia de innovaciones desde la investigación hacia la práctica institucional.

Investigaciones contemporáneas han abordado este problema desde diversas dimensiones. [20] y [21] proponen arquitecturas estructuradas en capas para gestionar la complejidad, logrando mejoras en la satisfacción del usuario, aunque persistiendo en diseños que dificultan la evolución no intrusiva. Por otro lado, la fragmentación entre lo pedagógico y lo técnico, descrita por [3] y [1], subraya que el problema no es solo la falta de algoritmos, sino la carencia de marcos que traduzcan principios educativos en restricciones arquitectónicas verificables.

2.4.2. Metodología de Análisis Comparativo

Para evaluar la efectividad y aplicabilidad de las propuestas existentes, se adoptó un enfoque sistemático basado en criterios de evaluación de la ingeniería de software, alineado con metodologías como el *Architecture Tradeoff Analysis Method* (ATAM) [43] y el estándar ISO/IEC/IEEE 42010:2022 [30]. Este marco metodológico permite una comparación objetiva entre estudios que utilizan enfoques similares pero contextos diversos.

Se definieron seis dimensiones de análisis que guardan relación directa con las propiedades PALS-A de esta tesis: escalabilidad, flexibilidad, mantenibilidad, interoperabilidad, rendimiento y validación empírica. La elección de estas dimensiones se fundamenta en trabajos previos que han validado su relevancia para arquitecturas de referencia [9, 10]. La comparación no solo busca listar funcionalidades, sino evaluar críticamente cómo la aplicación de patrones específicos (como microservicios o inyección de dependencias) impacta en la capacidad del sistema para evolucionar de manera sostenible.

2.4.3. Análisis de Propuestas y Síntesis de Hallazgos

El estudio revela una trayectoria evolutiva dividida en tres grandes paradigmas:

1. Arquitecturas Basadas en Componentes y Capas: Propuestas como las de [20] y [22] utilizan modelos jerárquicos (presentación, lógica, datos). Su principal hallazgo es que el aislamiento funcional mejora la mantenibilidad, pero su limitación reside en el acoplamiento con el núcleo del LMS, lo que genera una alta rigidez ante cambios en las políticas pedagógicas.

2. Descentralización mediante SOA y Microservicios: Para abordar la interoperabilidad, trabajos como [95] y [96] proponen servicios desacoplados. [96] reporta mejoras significativas en tiempos de respuesta mediante *API Gateways*, mientras que [97] formaliza la transición mediante el *Strangler Fig Pattern*, metodología que esta tesis adopta para garantizar la continuidad operativa.

3. Ecosistemas Cloud-Native e Inteligencia Artificial: La resiliencia y escalabilidad masiva son exploradas por [98] y [112]. En este nivel, la IA actúa como motor de personalización profunda; [113] demuestra una precisión del 87

La Tabla 2.1 sintetiza esta comparativa, destacando que si bien existe una convergencia hacia la modularidad técnica, la validación del impacto pedagógico real sigue siendo una cuestión pendiente en la mayoría de las implementaciones distribuidas.

Tabla 2.1: Matriz de Evaluación Comparativa de Trabajos Relacionados

Trabajo	Esc.	Flex.	Mant.	Inter.	Rend.	Valid.
Vagale et al. [20]	Medio	Bajo	Medio	Bajo	Alto	Alto
Musumba & Wario [95]	Alto	Alto	Medio	Alto	Medio	Bajo
Tsilingeridis et al. [96]	Alto	Alto	Alto	Alto	Alto	Alto
Rejina & Dwivedi [98]	Alto	Medio	Alto	Alto	Alto	Medio
Amala et al. [113]	Medio	Alto	Medio	Medio	Alto	Alto
Sutarman et al. [114]	Alto	Alto	Alto	Alto	Medio	Medio

2.4.4. Identificación de Brechas y Oportunidades: Hacia ARALS

A pesar de la madurez en microservicios, el análisis crítico identifica limitaciones estructurales: (1) ausencia de marcos unificados que integren la técnica con la pedagogía (el 'artefacto puente'); (2) falta de mecanismos de auto-evolución dinámica como bucles MAPE-K operativos; y (3) insuficiente validación que equilibre eficiencia técnica con gobernanza ética.

Estas brechas coinciden con las dificultades observadas en investigaciones previas

del autor sobre la incorporación de IA en entornos complejos [59, 60], donde la mera adición de algoritmos resultó insuficiente sin un diseño arquitectónico que explicitara responsabilidades y puntos de extensión. Asimismo, la falta de modelos que integren las dimensiones pedagógicas y éticas en la educación superior [61] refuerza la necesidad de ARALS. Como respaldo de esta solidez, la propuesta central ha sido aceptada para su publicación en la conferencia CIBSE 2026 [62], confirmando su relevancia frente a los desafíos de la Educación 4.0.

2.5. Fundamentos para la Construcción de la Arquitectura de Referencia

El presente capítulo ha establecido un andamiaje conceptual y metodológico para comprender la evolución, el estado actual y las limitaciones de las arquitecturas de software aplicadas a sistemas de e-learning adaptativos. A partir de la revisión sistemática de la literatura y del análisis comparativo de diferentes enfoques, se ha evidenciado una transición sostenida desde arquitecturas monolíticas hacia soluciones modulares, distribuidas y orientadas a servicios, fuertemente apoyadas en paradigmas *cloud-native*, microservicios e integración de técnicas de inteligencia artificial. Este desplazamiento se encuentra motivado por la necesidad de responder a requisitos crecientes de escalabilidad, flexibilidad, interoperabilidad y adaptación dinámica en entornos educativos digitales.

Simultáneamente, el estudio ha puesto de manifiesto la falta de marcos unificadores que articulen de manera sistemática los distintos paradigmas arquitectónicos empleados en sistemas adaptativos. Si bien se observan contribuciones valiosas en términos de patrones, estilos y tecnologías emergentes, todavía persiste una fragmentación conceptual que dificulta la comparación rigurosa entre propuestas, la transferencia de buenas prácticas y la consolidación de criterios de diseño compartidos. En este contexto, la convergencia entre adaptabilidad estructural, escalabilidad operativa y validación empírica se presenta como un desafío central aún no resuelto de manera integral.

En síntesis, las tendencias identificadas describen un escenario en el que la capacidad de evolución arquitectónica, la personalización dinámica y la eficiencia en el uso de recursos se han convertido en ejes vertebradores del diseño de sistemas de e-learning adaptativos. No obstante, las brechas detectadas ponen de relieve la necesidad de arquitecturas de referencia que integren explícitamente principios de auto-adaptación, mecanismos de evolución controlada y esquemas de evaluación sistemática de atri-

butos de calidad. Sobre esta base, el modelo arquitectónico propuesto en esta tesis —ARALS— se concibe como una respuesta estructurada a estas demandas, orientado a ofrecer una arquitectura unificada, coherente y empíricamente validable para la gestión del aprendizaje adaptativo. El capítulo siguiente desarrolla en detalle esta propuesta, explicitando sus principios de diseño, su estructura conceptual y los mecanismos que habilitan su carácter flexible, extensible y alineado con los requisitos de los sistemas educativos contemporáneos.

2.5.1. Contribuciones Teóricas

Desde una perspectiva teórica, este capítulo ha contribuido a la construcción de un marco conceptual integrado que articula de manera coherente los fundamentos de la arquitectura de software, las teorías de aprendizaje adaptativo y los principios de los sistemas complejos. Este marco permite interpretar los sistemas de e-learning adaptativos como sistemas software auto-organizativos, sometidos a restricciones arquitectónicas y pedagógicas, y proporciona el vocabulario y las estructuras conceptuales necesarias para su análisis sistemático.

Asimismo, se ha propuesto una taxonomía de patrones arquitectónicos aplicables a sistemas adaptativos, que organiza de manera estructurada las diversas aproximaciones encontradas en la literatura —incluyendo arquitecturas en capas, orientadas a servicios, de microservicios, dirigidas por eventos y *cloud-native*—. Esta clasificación no solo facilita la comparación entre soluciones, sino que también establece un punto de partida formal para la selección y combinación de patrones de diseño en función de requisitos específicos.

Finalmente, se han identificado y sistematizado principios de diseño arquitectónico específicos para sistemas que requieren adaptabilidad dinámica. Entre ellos destacan la modularidad y el desacoplamiento, la separación entre lógica funcional y lógica de adaptación, la externalización de configuraciones, y la observabilidad como condición necesaria para la auto-adaptación. Estos principios constituyen la base conceptual sobre la cual se construye la propuesta ARALS y orientan las decisiones de diseño expuestas en los capítulos posteriores.

2.5.2. Hallazgos Empíricos

Desde el punto de vista empírico, el capítulo ha permitido sintetizar un conjunto de tendencias verificables en la práctica actual de diseño de sistemas de e-learning adaptativos. En primer lugar, se ha observado una convergencia hacia arquitecturas

modulares y distribuidas, donde los microservicios y las soluciones *cloud-native* se consolidan como opciones preferentes para dar soporte a requisitos de alta disponibilidad, escalabilidad horizontal y despliegue continuo.

En segundo lugar, se ha constatado la integración generalizada de técnicas de aprendizaje automático y otras formas de inteligencia artificial como componentes centrales de los sistemas adaptativos. Estos mecanismos se utilizan tanto para la predicción del desempeño estudiantil como para la recomendación personalizada de contenidos, la detección temprana de riesgos y la optimización de trayectorias de aprendizaje.

En tercer lugar, se ha verificado un énfasis creciente en la adopción de estándares de interoperabilidad educativa, tales como SCORM, xAPI y LTI, que facilitan la portabilidad de contenidos, la integración de herramientas externas y la consolidación de datos de aprendizaje procedentes de múltiples fuentes. En conjunto, estos hallazgos empíricos delimitan el contexto tecnológico en el que debe inscribirse cualquier nueva propuesta arquitectónica que pretenda ser relevante y sostenible.

2.5.3. Direcciones Futuras

Las evidencias teóricas y empíricas recopiladas en este capítulo definen con claridad las direcciones futuras que orientan el resto de esta tesis. En primer lugar, se establece la necesidad de definir requisitos específicos para arquitecturas adaptativas escalables, que integren explícitamente atributos de calidad como escalabilidad, flexibilidad, interoperabilidad y capacidad de auto-adaptación. Estos requisitos sirven de base para la formulación de criterios de diseño y para la evaluación posterior de la propuesta arquitectónica.

En segundo lugar, se plantea el diseño de un framework arquitectónico unificado que recoja las mejores prácticas identificadas en la literatura y las articule en una estructura coherente, capaz de incorporar distintos patrones y estilos según las necesidades del dominio. Este framework se materializa en la arquitectura de referencia ARALS, cuya especificación detallada se desarrolla en los capítulos siguientes.

En tercer lugar, se proyecta la implementación y validación experimental de la propuesta, incluyendo la construcción de prototipos o sistemas de referencia que permitan evaluar empíricamente el comportamiento de la arquitectura bajo escenarios representativos. Este paso es crucial para confrontar el modelo teórico con condiciones reales de operación y para medir su impacto en términos de atributos de calidad arquitectónica y resultados educativos.

Finalmente, se prevé una evaluación comparativa con enfoques existentes, con el

fin de situar la propuesta en el contexto del estado del arte y de cuantificar sus aportes relativos. De este modo, el capítulo no sólo proporciona la fundamentación teórica y empírica necesaria, sino que también traza la hoja de ruta que guiará el desarrollo y la validación de soluciones arquitectónicas innovadoras orientadas a superar las limitaciones actuales en el dominio de los sistemas de e-learning adaptativos.

Capítulo 3

Fundamentos Metodológicos para la Transformación Arquitectónica hacia ARALS

3.1. Introducción

El presente capítulo constituye el núcleo metodológico de la investigación y se define estratégicamente como el marco preparatorio fundamental —o «pase de gol»— para la construcción de la arquitectura ARALS. El propósito central no es la formalización definitiva del modelo, sino la ejecución de un proceso sistemático de abstracción y síntesis que permita transitar, con rigor científico, desde la rigidez de los sistemas actuales hacia la flexibilidad de los entornos adaptativos.

Para asegurar pasos firmes en este recorrido, la estrategia se sustenta en tres pilares decisionales: el análisis de las plataformas LMS comerciales como punto de partida, la satisfacción de las propiedades adaptativas como meta operativa, y el uso de los «puentes conceptuales» (patrones, estilos y brechas) identificados en el estudio del estado del arte del Capítulo 2. Este camino no es fortuito; representa una decisión deliberada de utilizar el conocimiento acumulado en la ingeniería de software educativa para edificar un marco analítico que garantice la trazabilidad técnica absoluta entre los sistemas legados y la propuesta adaptativa final.

Este proceso de transformación, que se ilustra en la [Figure 3.1](#), constituye en sí mismo uno de los aportes sustanciales de esta tesis. Al definir una metodología re-

plicable para la evolución arquitectónica, se ofrece una solución al problema de la «rigidez estructural» identificado previamente. El proceso se articula en cuatro fases interconectadas:

En la **primera fase**, se determinó el núcleo funcional invariante del dominio mediante el análisis comparativo de cinco plataformas representativas y un catálogo de 29 patrones arquitectónicos. Este estudio permitió consolidar los componentes y relaciones esenciales que la comunidad científica reconoce como requisitos fundamentales para cualquier sistema de gestión del aprendizaje contemporáneo.

La **segunda fase** consistió en la *generalización* de estos hallazgos en un modelo abstracto estratificado por capas. El resultado se denomina *ARLMS (Arquitectura de Referencia para Sistemas de Gestión de Aprendizaje)*, un marco conceptual que vincula patrones, componentes y funcionalidades bajo principios de modularidad e independencia tecnológica, funcionando como la base estructural para las extensiones posteriores.

En la **tercera fase**, el modelo ARLMS se extiende hacia el dominio ALS (*Adaptive Learning Systems*). El objetivo central es facilitar la integración del ciclo de control autónomico *Monitor-Analyze-Plan-Execute over a shared Knowledge base (MAPE-K)* [34, 115, 116]. Para ello, se incorporaron componentes mediadores —motores de reglas, perfiladores y motores de recomendación— y se realizó un estudio de compatibilidad entre patrones tradicionales y patrones adaptativos (*Rule-Based, Event-Driven, Reflection, Blackboard*), sentando las bases de la auto-adaptación sistémica.

Finalmente, la **cuarta fase** sintetiza los resultados en una **estructura de transición** denominada **Arquitectura General Unificada (ARALS-Preliminar)**. Este esquema no constituye la definición final, sino un artefacto intermedio validado que articula el estudio comparativo con las extensiones adaptativas, sirviendo como el punto de partida directo para la definición formal de ARALS en el capítulo siguiente.

3.2. Fase 1: Caracterización de arquitecturas LMS

La definición de la arquitectura ARALS requiere identificar las invariantes estructurales que sustentan el dominio de gestión del aprendizaje. Mediante un proceso de ingeniería inversa y revisión técnica de cinco plataformas líderes (*Moodle, Open edX, Sakai, Chamilo y Claroline*), se relevaron los componentes recurrentes y estilos de diseño que rigen el e-learning a escala institucional [47, 37, 3]. La selección de estos casos respondió a criterios de madurez operativa y diversidad de estilos (monolíticos, SOA y modulares), permitiendo abstraer regularidades arquitectónicas que fundamentan el

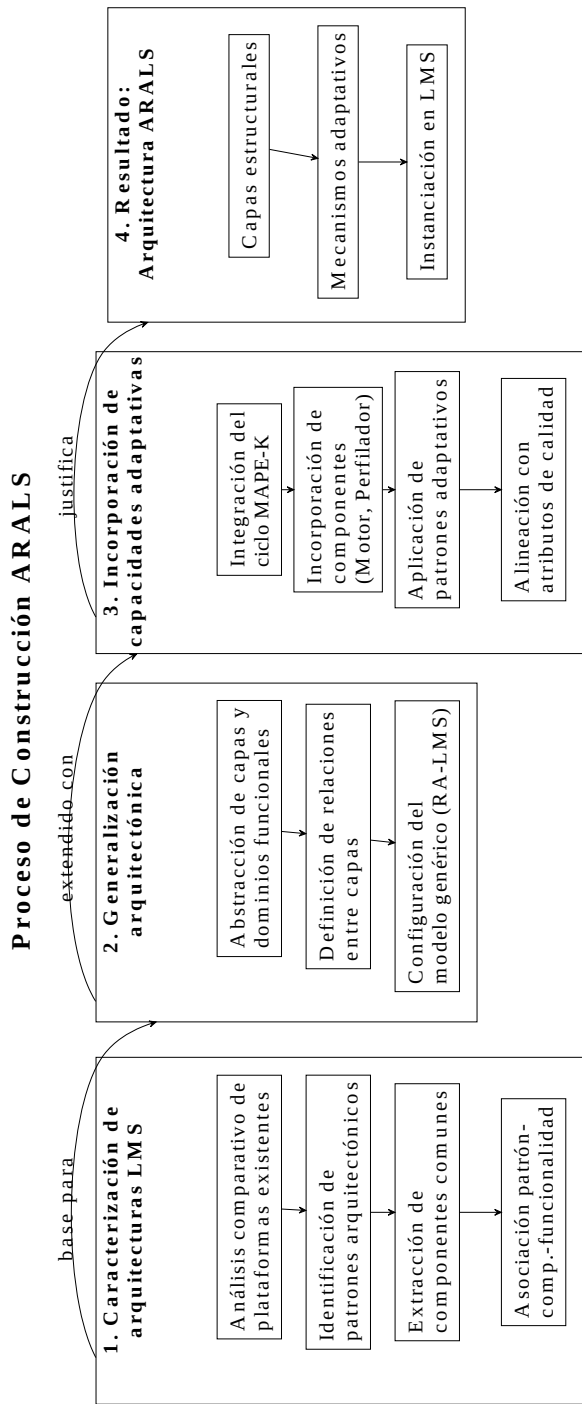


Figura 3.1: Fases del proceso analítico y preparatorio para la construcción de la Arquitectura de Referencia ARALS.

núcleo funcional de la propuesta [117, 4].

3.2.1. Núcleo Funcional y Estructuras Invariantes

El análisis sistemático permitió categorizar los componentes en siete dimensiones que definen el núcleo mínimo de un LMS. El **gestor de cursos, usuarios y roles** constituye el motor transaccional, apoyado por subsistemas de **evaluación, repositorios de contenido** y herramientas de **comunicación síncrona/asíncrona**. Estos elementos, presentes transversalmente, se complementan con **mecanismos de analítica básica** y soporte a estándares de interoperabilidad (*SCORM, IMS, LTI*). No obstante, la evidencia empírica indica que, si bien estas plataformas son robustas en gestión, presentan una marcada rigidez estructural” (*architectural stiffness*): menos del 15 % de las soluciones comerciales analizadas integran mecanismos nativos de adaptatividad profunda, limitándose mayoritariamente a una gestión estática bajo el paradigma *one-size-fits-all* [9, 118].

3.2.2. Análisis de Patrones y Estilos Arquitectónicos

Para profundizar en la mecánica de estas plataformas, se contrastaron sus estructuras con la taxonomía de 29 patrones de Farshidi et al. (2020). Los resultados, sintetizados en la [Table 3.1](#), revelan que los LMS tradicionales (*Moodle, Chamilo, Claroline*) se sustentan en una arquitectura **monolítica en capas** basada en **Client-Server** y **MVC**. En contraste, plataformas como *Open edX* han evolucionado hacia la **composición de servicios** (SOA/Microservicios), logrando mejoras de hasta un 300 % en tiempos de respuesta bajo carga alta [96]. Un hallazgo crítico es la presencia universal del patrón **Microkernel (Plug-in System)**, el cual otorga la extensibilidad necesaria para inyectar capacidades adaptativas sin alterar el núcleo transaccional. Sin embargo, patrones esenciales para la inteligencia dinámica como *Rule-Based System, Reflection* y *Blackboard* se encuentran ausentes, evidenciando el vacío de diseño que ARALS busca cubrir.

Tabla 3.1: Síntesis de patrones arquitectónicos y funcionalidades en LMS

Patrón	Componentes	Funcionalidad Invariante
Layers	Capas horizontales	Organización lógica de la gestión de cursos y datos.
MVC	Modelo-Vista-Ctrl	Separación de entidades educativas y visualización.
Microkernel	Núcleo y Plugins	Base de extensibilidad para actividades externas.
SOA / Microservices	API Gateway	Escalabilidad y despliegue independiente (Open edX).

Fase 1: Caracterización de arquitecturas LMS

Shared Repository	BD Centralizada	Consistencia de logs y calificaciones transaccionales.
Rule-Based	Motor de reglas	<i>Propuesto</i> : Personalización de rutas de aprendizaje.

Para profundizar en la caracterización funcional del dominio, la Tabla 3.2 sistematiza los componentes identificados en las cinco plataformas de referencia. Esta matriz permite discriminar entre el **núcleo invariante** (componentes presentes en el 100 % de los casos) y los **puntos de variabilidad diferencial**, fundamentales para identificar dónde se inyectará la lógica adaptativa. Se observa que, mientras la gestión educativa y la persistencia relacional son uniformes, la extensibilidad y la analítica avanzada aparecen como capacidades heterogéneas, dependientes de arquitecturas de *plugins* o *frameworks* complementarios.

Tabla 3.2: Matriz de componentes arquitectónicos identificados en plataformas LMS

Capa	Componente	Moodle	Open edX	Sakai	Chamilo	Claroline
<i>Presentación</i>	Temas/Plantillas	✓	✓	✓	✓	✓
	Interfaz responsiva	✓	✓	✓	✓	✓
<i>Lógica</i>	Gestor de cursos	✓	✓	✓	✓	✓
	Usuarios y roles	✓	✓	✓	✓	✓
	Evaluación y calificación	✓	✓	✓	✓	✓
	Repositorio contenidos	✓	✓	✓	✓	✓
	Analítica/Seguimiento	✓	✓	✓ ^D	✓	✓
	Portafolios	–	–	✓ ^D	–	–
<i>Servicios</i>	API REST / Web Services	✓	✓	✓	✓	✓
	Soporte LTI / SCORM	✓	✓	✓	✓	✓
	Microservicios	✓ ^P	✓	–	–	–
<i>Persistencia</i>	RDBMS (SQL)	✓	✓	✓	✓	✓
	NoSQL / LRS	–	✓	–	–	–

✓=Nativa; ✓^D=Derivada; ✓^P=Propuesta; –=Ausente.

La Figura 3.2 abstrae esta matriz en un modelo estratificado que representa la organización lógica predominante en el dominio. Esta vista es el «mapa de situación» que permite localizar los servicios nucleares y transversales (seguridad, *logging*, notificaciones) y sus puntos de integración con ecosistemas externos mediante la capa de servicios.

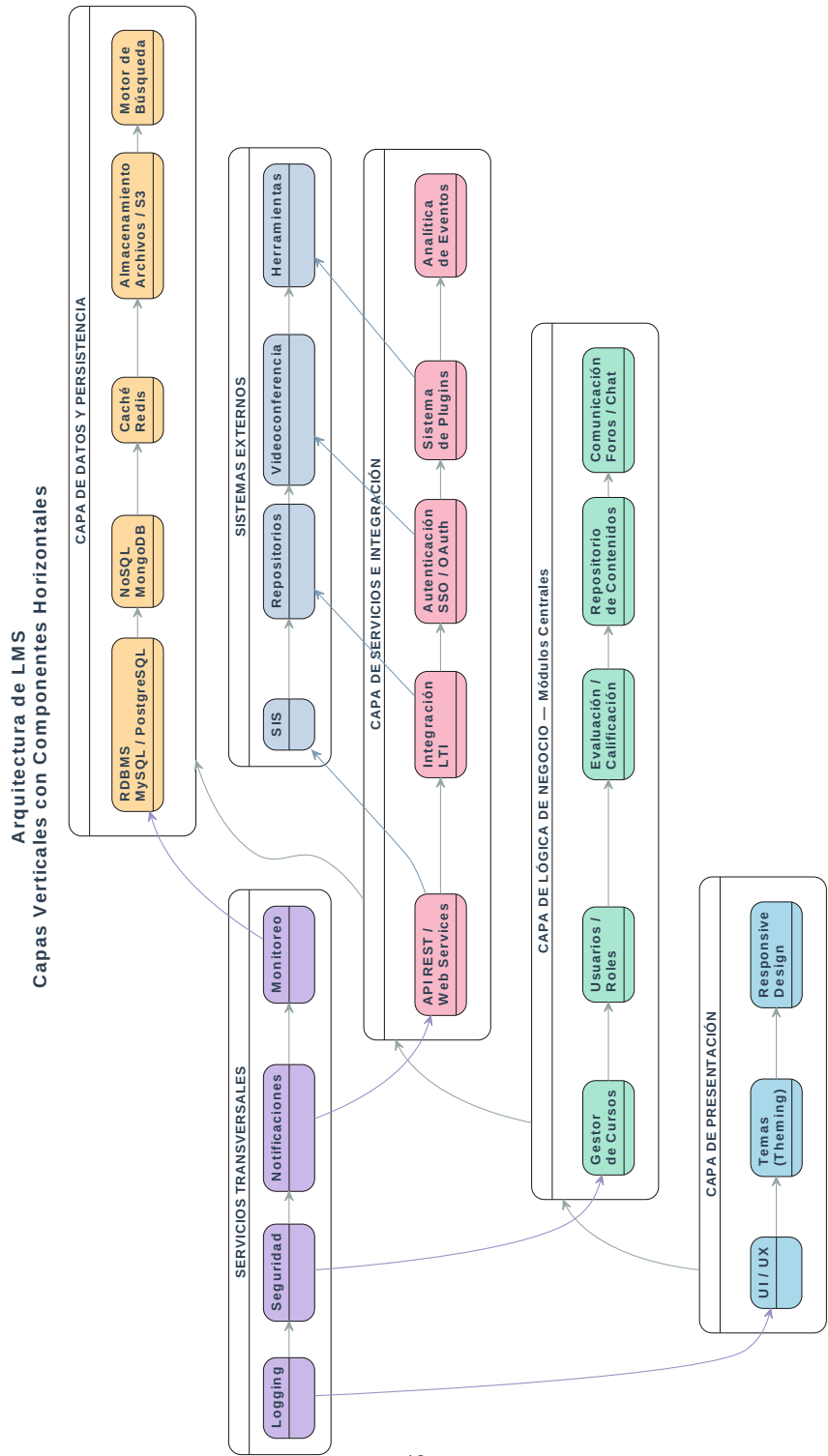


Figura 3.2: Modelo estratificado del dominio LMS: Organización de componentes funcionales e identificación de los puntos de integración para servicios transversales y adaptativos.

Fase 2: Generalización Arquitectónica

Como se desprende de la [Figure 3.2](#), la estructura se organiza en cuatro estratos horizontales: (i) **Presentación**, (ii) **Lógica de Negocio**, (iii) **Servicios e Integración** y (iv) **Datos**. Esta organización refleja el patrón predominante en arquitecturas maduras y constituye el escenario base sobre el cual se operará la transformación adaptativa. La relevancia de este modelo reside en que permite identificar los «puntos de anclaje» para el ciclo MAPE-K; específicamente, cómo la capa de Servicios e Integración funcionará como el puente para que los componentes especializados de ARALS orquesten la personalización sin comprometer la integridad de la Capa de Lógica de Negocio tradicional.

3.2.3. Implicaciones para el Diseño de ARALS

La caracterización de la Fase 1 establece las directrices para la arquitectura de referencia. Primero, ARALS debe heredar el **núcleo común validado** para asegurar compatibilidad con flujos tradicionales. Segundo, el patrón **Microkernel** fundamenta la decisión de estructurar la adaptatividad como un servicio desacoplado, evitando la reescritura total del código (*Conceptual Transformation Sequence - C1*). Tercero, la brecha detectada en estilos de procesamiento dinámico justifica la incorporación de una capa de **Adaptación Inteligente** basada en el ciclo **MAPE-K** [34]. Esta capa orquestrará componentes diferenciales como el motor de recomendación y el perfilador multidimensional, garantizando que el sistema evolucione de un repositorio estático a un entorno genuinamente adaptativo que cumpla con las propiedades **PALS** (*Properties of Adaptive Learning Systems*) [119, 120].

3.3. Fase 2: Generalización Arquitectónica

La segunda fase se orienta a la construcción de una abstracción de alto nivel que formalice las invariantes del dominio identificadas en la fase previa. El objetivo es transitar desde descripciones heterogéneas de plataformas específicas hacia un modelo conceptual uniforme e independiente de la tecnología, denominado **ARLMS (Arquitectura de Referencia para LMS)**, que funcione como el sustrato estructural para la posterior extensión adaptativa.

3.3.1. Estructura Estratificada y Relaciones de Control

Como se ilustra en la [Figure 3.3](#) y la [Figure 3.4](#), el modelo organiza los componentes en cuatro capas fundamentales —Interacción, Funcional, Integración y Datos— donde

los conectores son tratados como elementos de primera clase. Esta decisión metodológica es crítica: permite capturar los contratos de interacción y los flujos de eventos educativos que gobiernan el sistema, transformando la arquitectura de un conjunto estático de módulos a una red relacional de flujos de control.

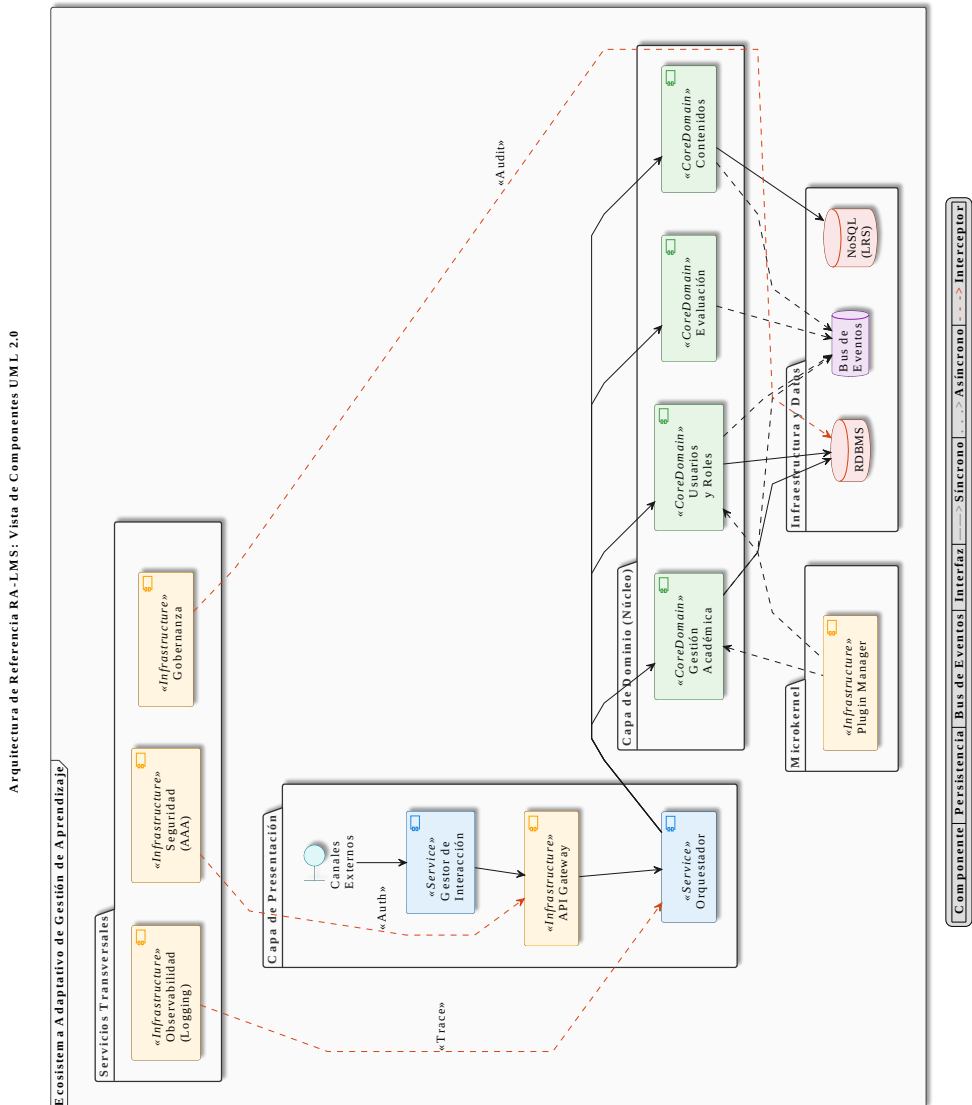


Figura 3.3: Abstracción de capas y dominios funcionales.

En este esquema, la **Capa de Interacción** abstrae los canales de acceso (Web, Mo-

ble, APIs), separando la experiencia de usuario de la lógica de negocio. Las solicitudes son mediadas por la **Capa de Integración**, que actúa como frontera arquitectónica mediante patrones *API Gateway*, garantizando el desacoplamiento y la gobernanza de interfaces. Por su parte, la **Capa Funcional** concentra los servicios nucleares (gestión académica, contenidos, evaluación), los cuales acceden a la **Capa de Datos** mediante relaciones de persistencia polígota. Un aspecto central es la identificación de **flujos de eventos educativos** asíncronos; el subsistema de analítica no solo consume datos, sino que produce eventos inmutables que habilitan la observabilidad longitudinal del estudiante, preparando el terreno para la futura integración de las propiedades PALS-A.

3.3.2. El Modelo RA-LMS: Invariantes y Extensibilidad

El proceso culmina en la configuración del modelo de referencia **RA-LMS** (Figura 3.5), una arquitectura canónica diseñada bajo el estilo *Layered Architecture* y los principios de *Domain-Driven Design* (DDD) [15, 121]. El modelo se descompone en cinco capas específicas que aseguran la separación de preocupaciones y la orquestación eficiente de casos de uso:

- a) **Presentación y Aplicación:** Implementan el patrón *Backend for Frontend* (BFF) y servicios de orquestación tipo *Facade*, centralizando la autenticación, el versionado de contratos y el *rate limiting* [40, 73].
- b) **Dominio:** Constituye el núcleo conceptual, segregando responsabilidades en subdominios de alta cohesión (Gestión Académica, Usuarios, Contenidos, Evaluación y Analítica). Este diseño previene la contaminación de la lógica educativa con preocupaciones de infraestructura.
- c) **Sistema de Extensibilidad:** Basado en el patrón **Microkernel** [122], formaliza la capacidad de incorporar módulos mediante *hooks* y *extension points*, permitiendo una personalización institucional sin alterar el código fuente (*Conceptual Transformation Sequence - C1*).
- d) **Integración y Persistencia:** Operan de manera complementaria mediante un **Bus de Eventos** (EDA) y persistencia polígota. Se utilizan RDBMS para datos transaccionales y *Learning Record Stores* (LRS) basados en xAPI para el registro inmutable de experiencias [24].

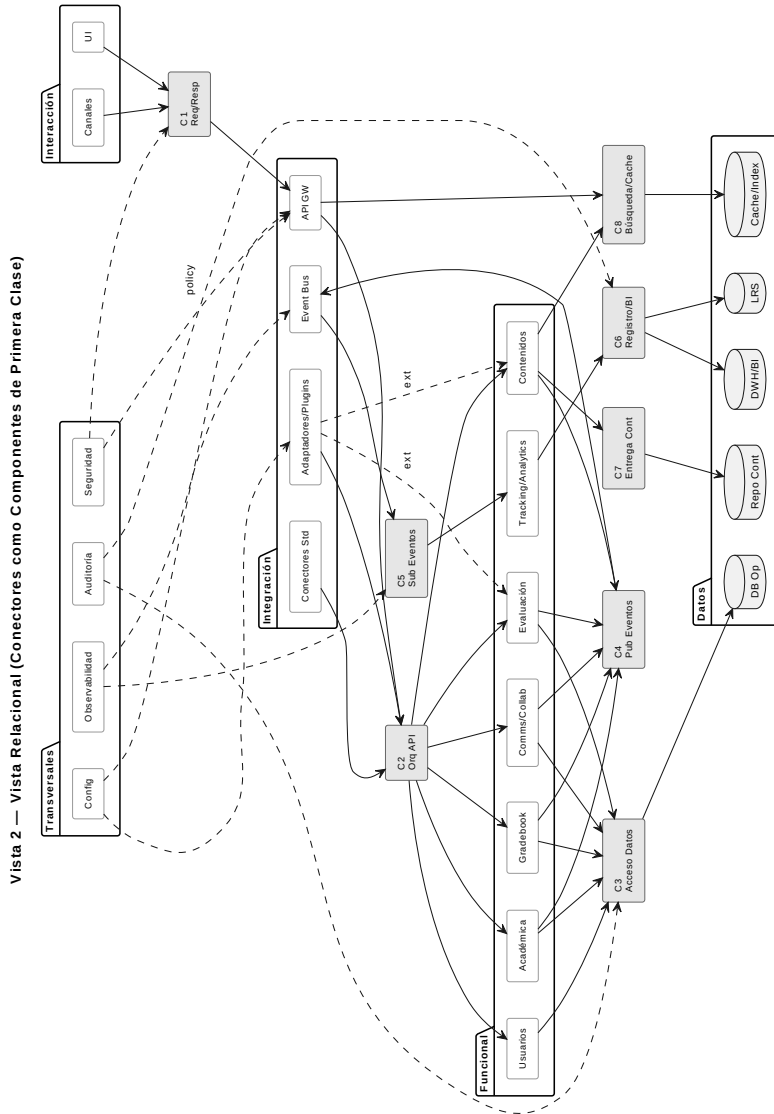


Figura 3.4: Vista relacional: los conectores como elementos de primera clase.

El RA-LMS actúa como una *arquitectura de transición* metodológica. Es lo suficientemente precisa para guiar la comparativa entre plataformas existentes y lo suficientemente abstracta para permitir la inyección del ciclo *MAPE-K* y los mecanismos de inferencia adaptativa. Al explicitar los puntos estructurales de variabilidad y tratar los servicios transversales (seguridad, observabilidad, auditoría) como aspectos de

primera clase, el modelo asegura que la evolución hacia ARALS no comprometa la estabilidad del núcleo transaccional. Esta fase cumple así con los criterios de validez para arquitecturas de referencia: definición clara del contexto de aplicación y un nivel de detalle que guía el diseño concreto sin prescribir tecnologías específicas [10].

3.4. Relación entre propiedades, componentes y arquitectura subyacente

La arquitectura RA-LMS constituye la base estructural necesaria para iniciar la transformación sistemática hacia entornos adaptativos. Sin embargo, para que el "pase de gol" hacia ARALS sea efectivo, es imperativo mapear la interacción entre los **componentes funcionales**, los **estilos arquitectónicos** que rigen su comunicación y las **propiedades adaptativas (PALS)** que definen a un ALS genuino. Esta trazabilidad permite aplicar la **Secuencia de Transformación Conceptual (C1)**, identificando qué elementos del núcleo tradicional son habilitadores de la personalización y cuáles actúan como cuellos de botella debido a la rigidez arquitectónica" propia de los sistemas legados.

La Tabla 3.3 sistematiza esta matriz de trazabilidad, vinculando cada estrato de RA-LMS con los estilos predominantes y las propiedades que facilitan. Por ejemplo, el uso del estilo *Microkernel* en el sistema de extensibilidad es el facilitador técnico de la **Escalabilidad** y la **Interoperabilidad**, permitiendo la inyección de motores de inferencia sin alterar el código base. De igual forma, la implementación de un *Shared Repository* en la capa de datos es la precondition para la **Personalización Adaptativa Colaborativa (PAC)**, al centralizar los modelos de usuario multidimensionales (cognitivos, emocionales y sociales). Esta caracterización asegura que ARALS no sea una propuesta ad-hoc, sino una evolución fundamentada que preserva la estabilidad operativa mientras inyecta inteligencia pedagógica.

Tabla 3.3: Matriz de trazabilidad: Componentes RA-LMS, Estilos y Propiedades PALS.

Componente	Estilos Predominantes	Propiedades PALS Soportadas
Capa Presentación	<i>MVC, PAC, Interceptor</i>	Cuestiones Cognitivas: Adaptación de densidad visual. Retroalimentación: Captura de eventos de interacción.
Capa Aplicación	<i>Broker, Facade, Gateway</i>	Escalabilidad: Control de carga y routing. Accountability: Auditoría de invocaciones y trazabilidad.

Fundamentos Metodológicos para la Transformación Arquitectónica hacia ARALS

Capa Dominio	<i>Rule-Based, Event-Based, Shared Repository</i>	PAC: Gestión de cohortes adaptativas. Evaluabilidad: Reglas dinámicas de ajuste de dificultad. Multistakeholder: Roles diferenciados.
Extensibilidad	<i>Microkernel, Reflection</i>	Escalabilidad/Interoperabilidad: Inyección de algoritmos heterogéneos mediante plugins dinámicos.
Integración	<i>Pub-Sub, EDA, Broker</i>	Retroalimentación Adaptativa: Reacción asíncrona en tiempo real. Accountability: Trazabilidad mediante xAPI/LTI.
Persistencia	<i>Polyglot Persistence, LRS</i>	PAC: Almacenamiento de modelos multidimensionales. Evaluabilidad: Análisis retrospectivo mediante Big Data.
Transversales	<i>AOP, Interceptor, Rule-Based</i>	Gobernanza y Compliance: Políticas de autorización y auditoría inmutable de decisiones algorítmicas.

El análisis de la matriz revela que, si bien RA-LMS ofrece una base sólida mediante el desacoplamiento de capas y el uso de microkernels, presenta carencias críticas para alcanzar el estado de ALS genuino. Se observa una ausencia sistemática de estilos tipo **Blackboard** para la resolución colaborativa de perfiles de usuario y, fundamentalmente, la falta de un ciclo de control autónomico **MAPE-K** integrado que orqueste la autogestión estructural. Estas brechas arquitectónicas —especialmente la rigidez en la evaluación dinámica de reglas pedagógicas— confirman que el modelo tradicional ha llegado a su límite de adaptatividad. Consecuentemente, el diseño de ****ARALS**** en el capítulo siguiente se presenta como la solución necesaria para inyectar estos mecanismos de inferencia y metaprogramación, operacionalizando la transformación LMS→ALS de manera no intrusiva.

3.5. Fase 3: Incorporación de capacidades adaptativas

Esta fase representa el punto de inflexión metodológico al desplazar el foco desde la generalización estructural del LMS hacia la inyección de inteligencia sistémica. El tránsito desde una arquitectura de gestión hacia una orientada a la *adaptación del aprendizaje* no se propone como una reingeniería disruptiva, sino como una extensión fundamentada del modelo RA-LMS. Este proceso permite que la adaptabilidad se incorpore como una propiedad estructural y no como un añadido ad-hoc, vinculando

las decisiones de diseño con los principios de los sistemas autoadaptativos y el ciclo de control autónomo *MAPE-K* (*Monitor-Analyze-Plan-Execute-Knowledge*) [34, 115].

3.5.1. Propiedades Adaptativas PALS-A: Atributos Arquitectónicamente Influenciables

La evolución hacia los *Adaptive Learning Systems* (ALS) exige que la arquitectura soporte modificaciones de comportamiento en tiempo de ejecución. Para operacionalizar este requisito, se definen las **PALS-A** (*Properties for Adaptive Learning Systems - Architecturally influenceable*), un conjunto de capacidades cuya realización efectiva depende críticamente del diseño estructural subyacente (modularidad, acoplamiento y estilos adoptados). Estas propiedades actúan como el puente entre las metas pedagógicas y la ingeniería de software, permitiendo que la adaptabilidad sea medida, evaluada y optimizada mediante patrones específicos.

Definition 3.1 (Propiedades PALS-A). Las **PALS-A** son los pilares de la personalización y eficiencia operativa de un ALS. Su viabilidad técnica reside en la configuración de los componentes y mecanismos de integración:

- **Personalización Adaptativa Contextualizada (PAC):** Ajuste dinámico de contenidos, interfaces y secuencias según el perfil y contexto del estudiante. Depende de la modularidad y el monitoreo en tiempo real.
- **Interoperabilidad:** Integración semánticamente coherente entre LMS, motores de recomendación y herramientas externas, favoreciendo la orquestación dinámica.
- **Escalabilidad en Edge-Computing:** Ejecución de adaptaciones en nodos distribuidos o clientes para reducir latencia y asegurar personalización *offline* mediante arquitecturas *lightweight*.
- **Retroalimentación Adaptativa Continua:** Respuesta inmediata a las acciones del usuario mediante bucles de *feedback* pedagógico procesados en tiempo real.
- **Multistakeholder Personalization:** Capacidad de ofrecer vistas y servicios diferenciados por rol (estudiante, docente, administrador) mediante control de acceso contextual.
- **Adaptación Cognitivo-Afectiva:** Representación y ajuste de la experiencia según la carga cognitiva, estilo de aprendizaje y estado emocional detectado.
- **Accountability y Gobernanza:** Trazabilidad y explicabilidad de las decisiones adaptativas mediante mecanismos de transparencia algorítmica y auditoría de rutas.
- **Evaluación Adaptativa:** Ajuste dinámico de estrategias y criterios de evaluación según el progreso, requiriendo motores de rúbricas desacoplados.

3.5.2. Mapeo de Propiedades y Estilos Arquitectónicos

La materialización de las PALS-A requiere la selección estratégica de patrones que influyan positivamente en los atributos de calidad del sistema. La [Table 3.4](#) sintetiza esta relación, justificando cómo estilos como *Blackboard*, *Rule-Based Systems* o *Event-Driven Architectures* actúan como habilitadores de la inteligencia adaptativa [123].

Tabla 3.4: Relación entre propiedades PALS-A y estilos arquitectónicos habilitadores.

Propiedad PALS-A	Arquitecturas Relevantes	Justificación Técnica
PAC	Microservicios, SOA, Reflection	Favorecen modificabilidad y flexibilidad para configuraciones en tiempo de ejecución.
Interoperabilidad	SOA, Broker, Pipes and Filters	Promueven integrabilidad y modularidad entre sistemas distribuidos.
Escalabilidad Edge	Peer-to-Peer, Space-Based	Optimizan la eficiencia del desempeño en entornos de carga dinámica.
Retroalimentación	Blackboard, Pub-Sub, Microkernel	Facilitan la <i>responsiveness</i> necesaria para bucles de <i>feedback</i> dinámico.
Cognitivo-Afectiva	Rule-Based, Interpreter, IA-Microservices	Patrones para dominios que requieren razonamiento, inferencia y sistemas inteligentes.
Gobernanza/Eval.	Layered, Blackboard, Microkernel	Garantizan trazabilidad (<i>traceability</i>) y toma de decisiones en <i>pipelines</i> de evaluación.

3.6. Fase 4: Arquitectura General Unificada

La Arquitectura General Unificada constituye la síntesis conceptual de las fases previas, integrando las invariantes estructurales de RA-LMS con las propiedades adaptativas PALS-A. Su propósito es establecer un artefacto de transición que, siguiendo el enfoque *Views & Beyond* de Clements et al. [37], organice la complejidad sistémica mediante capas y una dimensión transversal centrada en los *Tipos de Componentes Adaptativos (ALS)*. Este modelo preliminar no es una definición cerrada, sino el marco estructural que fundamenta la inyección del ciclo MAPE-K en el capítulo siguiente.

3.6.1. Estructura Integrada por Capas y Estilos

La arquitectura se articula en cinco estratos que aplican principios de *Clean Architecture* y *Domain-Driven Design* (DDD). La **Capa de Interacción** (Web, Mobile, API) utiliza estilos *Client-Server* y *PAC* para capturar eventos de usuario que alimentarán el motor de adaptación. Estos flujos son gestionados por la **Capa de**

Fase 4: Arquitectura General Unificada

Aplicación, donde un *API Gateway* y un *Orquestador de Servicios* (*Broker*, *Facade*) centralizan el control de acceso y la compensación de operaciones complejas [62, 73].

En el núcleo, la **Capa Funcional (Dominio)** encapsula los seis servicios educativos esenciales (Gestión Académica, Usuarios, Contenidos, Evaluación, Comunicación y Analítica), elevando la **Observación y Analítica** a componente de primera clase mediante estilos *Event-Based* y *Batch Sequential*. Esta lógica se apoya en una **Capa de Integración** que implementa estándares (LTI, xAPI, QTI) y un **Bus de Eventos** (EDA) para asegurar el desacoplamiento temporal de los componentes. Finalmente, la **Capa de Persistencia** aplica el patrón *Polyglot Persistence* [71], segregando datos transaccionales de las trazas inmutables en el *Learning Record Store* (LRS), mientras que el **Sistema de Extensibilidad** (*Microkernel*) habilita la carga dinámica de plugins adaptativos mediante *Reflection* [122].

Como puente hacia ARALS, el modelo identifica cinco tipos de componentes adaptativos relacionados con PALS-A, representadas en la [Figure 3.6](#), que actúan como metatipos para clasificar la funcionalidad de cada componente en el proceso de personalización:

- T1. **Captura y Observación:** Articula el controlador de presentación y el LRS para instrumentar el comportamiento (*Monitor*).
- T2. **Modelado del Aprendiz:** Vincula la gestión de usuarios y el Data Warehouse para la construcción de perfiles multidimensionales (*Analyze*).
- T3. **Adaptación Pedagógica:** Orquesta la lógica de contenidos y evaluaciones mediante motores *Rule-Based* (*Plan*).
- T4. **Adaptación de Interfaz:** Ejecuta ajustes visuales y estructurales dinámicos (*Execute*).
- T5. **Gestión del Conocimiento:** Consolida las reglas, ontologías y metadatos que guían la inteligencia sistémica (*Knowledge*).

3.6.2. Síntesis y Transición hacia ARALS

La Arquitectura General Unificada culmina el proceso preparatorio al identificar los puntos de variabilidad donde RA-LMS se transformará en ARALS. Al explicitar la influencia de las capacidades adaptativas sobre los componentes base, este modelo permite que la transición hacia la arquitectura definitiva no sea una reingeniería ciega, sino un diseño dirigido por calidad [15].

Consecuentemente, el Capítulo 4 operacionalizará esta estructura mediante la incorporación formal del ciclo *MAPE-K*, motores de inferencia basados en *Interpreter*,

y modelos de usuario estructurados bajo el patrón *Blackboard*. Este 'pase de gol' asegura que *ARALS* cuente con un sustento empírico y conceptual robusto, capaz de evolucionar sistemas legados hacia entornos de aprendizaje inteligentes, trazables y escalables, cumpliendo así con los objetivos de transformación sistemática planteados en esta tesis.



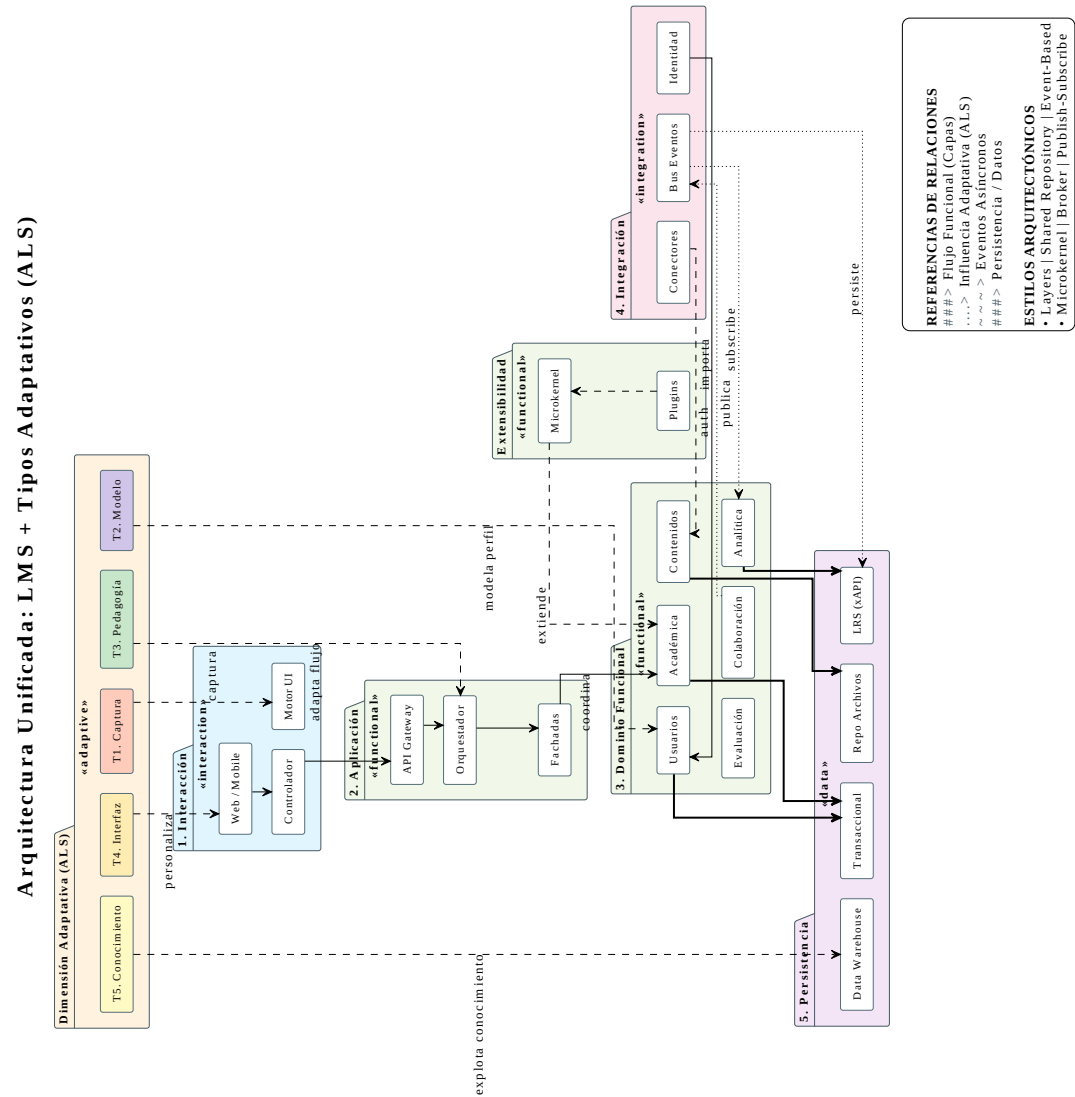


Figura 3.6: Arquitectura General Unificada (ARALS-Preliminar): Síntesis estructural que articula el núcleo transaccional con los metatipos adaptativos T1-T5 para la inyección del ciclo MAPE-K.

Capítulo 4

Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo

4.1. Introducción

El diseño de arquitecturas para Sistemas de Aprendizaje Adaptativo (ALS) trasciende la mera adopción de estructuras ingenieriles sólidas; exige una perspectiva integral que reconozca las particularidades del dominio educativo digital y la necesidad de una evolución operativa continua. Tal como se fundamentó en los capítulos precedentes, la construcción de una arquitectura de referencia para ALS debe articular simultáneamente tres pilares: (i) los invariantes arquitectónicos de los LMS consolidados, (ii) los patrones estructurales validados históricamente en el diseño educativo, y (iii) las propiedades adaptativas (PALS) cuya realización técnica es indisociable de las decisiones de diseño estructural.

Bajo esta premisa, el enfoque adoptado en esta investigación se alinea con los lineamientos del diseño guiado por calidad y el análisis estructural de software. El objetivo es ofrecer un marco conceptual y técnico capaz de sustentar arquitecturas robustas que, lejos de ser réplicas de modelos existentes, se fundamenten en evidencia

empírica y mecanismos formales de análisis. Esta articulación metodológica permite razonar sobre la coherencia interna del sistema y garantiza que la solución propuesta responda a los requisitos específicos de los entornos de aprendizaje inteligente.

Consecuentemente, este capítulo presenta la Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo (ARALS), concebida como una estructura extensible que integra hallazgos empíricos previos con los fundamentos teóricos de la ingeniería de software. En oposición a concepciones prescriptivas o monolíticas, ARALS se define como un marco orientado a guiar la evolución de plataformas de e-learning tradicionales hacia ecosistemas adaptativos y escalables. Su diseño se sustenta en principios de modularidad, separación de responsabilidades y gestión de la variabilidad, conforme a los patrones y estilos arquitectónicos consolidados en la literatura especializada [54, 55].

La construcción de ARALS se articula mediante un proceso metodológico compuesto por cuatro fases complementarias, definidas para asegurar un tránsito organizado, desde la identificación de invariantes arquitectónicos hasta la validación estructural del modelo preliminar. Estas fases se encuentran alineadas con el marco metodológico presentado en el Capítulo 3 y responden a lineamientos aceptados en la ingeniería de software, tales como el diseño guiado por calidad, la abstracción progresiva y el razonamiento basado en vistas arquitectónicas. Cada fase contribuye de manera incremental a consolidar una propuesta arquitectónica coherente, justificable y fundamentada tanto en la literatura como en la evidencia derivada del análisis sistemático de plataformas LMS.

1. **Identificación de componentes comunes en arquitecturas de LMS:** Esta etapa inicial comprende un análisis comparativo sistemático de sistemas de gestión del aprendizaje (LMS), orientado a detectar componentes recurrentes, estilos arquitectónicos predominantes y puntos de variabilidad estructural. Se adopta una estrategia de recuperación de conocimiento arquitectónico (*Architectural Knowledge*) para extraer patrones que luego fundamentarán la configuración estructural de ARALS [124].
2. **Diseño de la arquitectura de referencia:** Basándose en los componentes identificados, se construye un modelo arquitectónico que aplica estilos consolidados —como la arquitectura por capas, el patrón *Service Access Layer* y mecanismos de adaptación— con el objetivo de establecer una plantilla reusable y modular. Este diseño promueve la separación de responsabilidades, el uso de contratos explícitos, y la incorporación de puntos de extensión bien definidos,

facilitando su evolución y adaptación a distintas plataformas [47, 25].

3. **Validación conceptual y verificaciones de consistencia:** La última fase evalúa la solidez estructural de ARALS mediante una validación conceptual que verifica su coherencia interna y su completitud frente a las propiedades adaptativas (PALS-A) y los hallazgos del análisis comparativo de LMS. Esta revisión sistemática, de carácter no empírico, asegura la alineación de capas y componentes con los mecanismos propuestos, preservando la trazabilidad y consistencia semántica desde el modelo preliminar del Capítulo 3. Asimismo, se analiza la factibilidad de instanciar la arquitectura en plataformas reales como Odoo eLearning mediante un estudio de compatibilidad argumentado, garantizando que el modelo sea conceptualmente robusto, general y adecuado como arquitectura de referencia para sistemas de aprendizaje adaptativo.

Como resultado, ARALS se posiciona como un marco arquitectónico de referencia con el potencial de guiar procesos de transformación evolutiva en plataformas educativas, facilitando la incorporación de capacidades adaptativas sin comprometer la compatibilidad ni la mantenibilidad de los sistemas existentes. Su diseño orientado a componentes modulares, interoperables y reconfigurables permite su aplicación progresiva, tanto en entornos académicos como corporativos, alineándose con la visión de ingeniería de software basada en patrones y reutilización estructurada [25, 54].

4.2. Diseño de la arquitectura de referencia

La arquitectura ARALS (Arquitectura de Referencia para Adaptive Learning Systems) surge como una síntesis analítica que formaliza las regularidades identificadas en plataformas líderes —Moodle, Open edX, Canvas LMS, Sakai y Chamilo— en un modelo jerárquico y trazable. Representada en la Figura 4.1, esta propuesta trasciende lo prescriptivo para ofrecer un marco de referencia robusto que sistematiza interacciones, flujos de datos y puntos de integración. Mediante un enfoque *pattern-driven*, ARALS consolida estilos de *Layers*, *Component-Based* y *SOA* como invariantes del dominio, estableciendo una hoja de ruta conceptual alineada con las Propiedades Adaptativas (PALS-A) detalladas en 3.1.

Estructuralmente, el modelo se organiza en seis capas interrelacionadas —*Presentación*, *Servicios*, *Adaptación Inteligente*, *Dominio/Modelo*, *Datos* e *Integraciones Externas*— que operacionalizan el ciclo adaptativo de captura, análisis, planificación, ejecución y registro. Esta estratificación asegura la mantenibilidad y el acoplamiento

débil del sistema, permitiendo que la arquitectura actúe como un artefacto de consolidación para la toma de decisiones tecnológicas fundamentadas en entornos educativos dinámicos.

La dinámica operativa se sustenta en trece relaciones (*Rel1–Rel13*) bajo el paradigma MAPE-K. Mientras *Rel1* y *Rel2* canalizan datos hacia los módulos analíticos, las relaciones *Rel3*, *Rel4* y *Rel5* garantizan la coherencia semántica y la interoperabilidad externa mediante estándares como LTI, xAPI y SCORM. El ciclo de ajuste en tiempo real se ejecuta de la *Rel6* a la *Rel10*, culminando en la actualización de perfiles, el registro de progreso y la integración de eventos analíticos a través de las relaciones *Rel11* a *Rel13*. Este modelo, en su estado de caracterización, fundamenta el análisis granular subsiguiente de cada capa para demostrar su adecuación a los principios de adaptabilidad de la investigación.

Tabla 4.1: Matriz de relaciones entre capas (*Rel1–Rel13*, versión compacta)

Relación	UI	SERV	ADAPT	DOM/DAT/EXT
Rel1	UI→S	EvUI→S	—	—
Rel2	—	S→A	A→S (propag.)	—
Rel3	—	—	A→Dmn	Dmn int.
Rel4	—	—	A→Dat	Dat pers.
Rel5	—	—	—	EXT int.
Rel6	Métr.→A	—	Capt.sen.	—
Rel7	—	S→A	DatApr→A	—
Rel8	—	—	ResA→Adec	—
Rel9	—	—	DecA→Exec	—
Rel10	AdjUI	—	Exec→UI	—
Rel11	—	—	—	ActPerf
Rel12	—	—	—	RegProg
Rel13	—	—	—	IntEvDat

La Tabla 4.1 debe interpretarse como una matriz que organiza, en filas, cada una de las relaciones identificadas en el modelo (*Rel1–Rel13*), y en columnas, las capas arquitectónicas involucradas en dichas interacciones. El contenido de cada celda corresponde a una sigla que indica el tipo de flujo o la función conceptual asociada a la relación cuando esta conecta dos capas específicas. De este modo, una celda vacía señala que la relación no involucra a esa capa, mientras que una sigla presente indica la naturaleza del intercambio entre los componentes que pertenecen a la capa correspondiente. Esta estructura permite visualizar rápidamente qué capas participan en

Diseño de la arquitectura de referencia

cada relación y cuál es el rol funcional que desempeñan dentro del ciclo adaptativo del sistema.

Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo

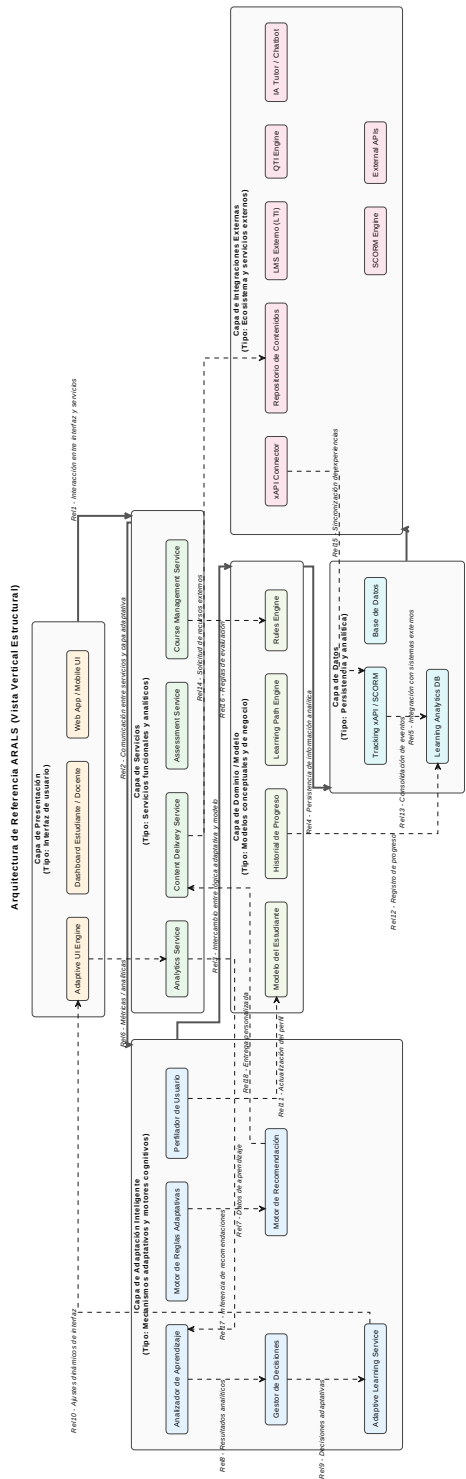


Figura 4.1: Arquitectura de Referencia ARALS: Representación estratificada en seis capas funcionales y su red de trece relaciones semánticas (Rel1-Rel13) para la gestión del ciclo adaptativo.

4.2.1. Estructura general y principios de diseño

La arquitectura ARALS se fundamenta en los principios de separación de responsabilidades, acoplamiento débil y extensibilidad progresiva, estructurándose mediante los patrones *Layered Architecture* y *Microkernel* para integrar funciones adaptativas sin alterar el núcleo del sistema. Esta organización, orientada a servicios (SOA) y gobernada por eventos, se despliega a través de una Capa de Presentación (UI) que actúa como punto de interacción principal mediante interfaces web, móviles y un *API Gateway*. En esta instancia, el *Adaptive UI Engine* dinamiza la experiencia del usuario aplicando reglas de personalización visual y funcional sensibles al contexto, asegurando una interfaz coherente bajo principios de diseño inclusivo.

Seguidamente, la Capa de Aplicación y Servicios concentra la lógica académica y administrativa mediante servicios desacoplados bajo los patrones *Service Layer* y *Component-Based*, facilitando la interoperabilidad y proveyendo soporte a la Capa de Adaptación Inteligente. En este núcleo estratégico se operacionaliza el ciclo *Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K)* a través de componentes como el *Motor de Recomendación*, el *Perfilador de Usuario* y el *Learner Model Engine*, los cuales permiten que el sistema evolucione dinámicamente y ajuste la trayectoria de aprendizaje según las necesidades detectadas en el aprendiz.

El soporte semántico de este proceso reside en la Capa de Lógica y Dominio, donde el *Learning Path Engine* garantiza la consistencia pedagógica de las adaptaciones. Por su parte, la Capa de Persistencia y Datos optimiza el rendimiento mediante patrones *Repository* y *CQRS*, integrando estándares de interoperabilidad como xAPI, SCORM y QTI para asegurar una visión unificada de la interacción. Finalmente, la Capa de Integraciones Externas actúa como un subsistema que, mediante conectores LTI y adaptadores específicos, vincula a ARALS con repositorios externos, sistemas de tutoría basados en IA y plataformas de analítica, consolidando un ecosistema educativo escalable en entornos tecnológicos heterogéneos.

La articulación de estas seis capas configura una estructura coherente y escalable que unifica los modelos tradicionales de LMS con capacidades adaptativas avanzadas. En ARALS, esta organización representa la convergencia entre patrones estructurales consolidados —*Layers*, *MVC*, *SOA* y *Microkernel*— y mecanismos específicos de adaptación —*Rule-Based System*, *Event-Driven*, *Reflection* y *Blackboard*—, permitiendo conciliar la estabilidad operativa con la flexibilidad requerida para la personalización pedagógica. Más que una disposición estratificada, ARALS se consolida como una arquitectura de referencia que formaliza el conocimiento derivado del estudio empírico

previo, ofreciendo un marco reutilizable y una guía metodológica para la toma de decisiones tecnológicas en escenarios de adaptación continua. Por consiguiente, el análisis granular desarrollado en las secciones posteriores profundiza en las responsabilidades, componentes y relaciones funcionales de cada capa, evidenciando su contribución individual al ciclo adaptativo y su alineación con los requisitos arquitectónicos establecidos en las fases previas de esta investigación.

4.2.2. Capa de Presentación – Interfaces y Experiencia del Usuario

La arquitectura ARALS se fundamenta en los principios de separación de responsabilidades, acoplamiento débil y extensibilidad progresiva, empleando los patrones *Layered Architecture* y *Microkernel* para integrar capacidades adaptativas sin comprometer el núcleo sistémico. Esta organización, orientada a servicios (SOA) y eventos, se despliega inicialmente en la Capa de Presentación (UI), donde el *API Gateway* y el *Adaptive UI Engine* orquestan una experiencia de usuario personalizada e inclusiva a través de interfaces web y móviles sensibles al contexto. Seguidamente, la Capa de Aplicación y Servicios centraliza la lógica académica mediante componentes desacoplados (*Service Layer* y *Component-Based*) que dan soporte a la Capa de Adaptación Inteligente. En este núcleo estratégico se operacionaliza el ciclo *Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K)* mediante el *Motor de Recomendación*, el *Perfilador de Usuario* y el *Learner Model Engine*, permitiendo ajustes dinámicos en la trayectoria de aprendizaje basados en las necesidades del aprendiz.

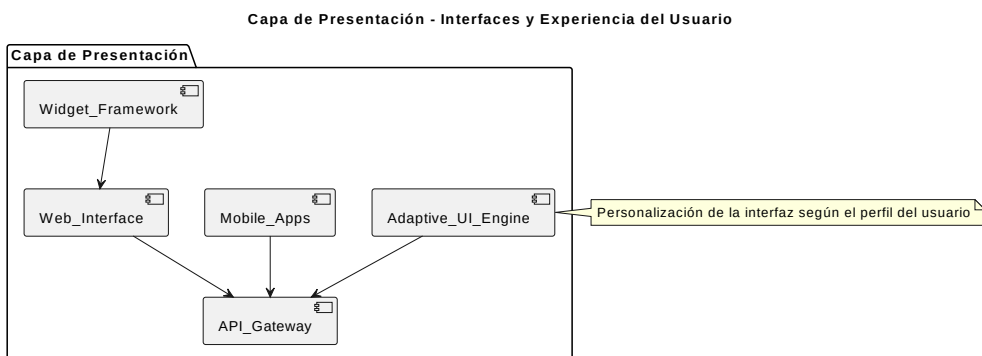


Figura 4.2: Capa de Presentación: Organización de componentes para la orquestación de la experiencia de usuario y la personalización dinámica de la interfaz mediante el *Adaptive UI Engine*.

El soporte semántico y operativo de este flujo se articula en las capas de Lógica y Dominio y de Persistencia y Datos. Mientras el *Learning Path Engine* asegura la coherencia pedagógica de las adaptaciones, el uso de los patrones *Repository* y *CQRS* bajo estándares xAPI, SCORM y QTI garantiza el rendimiento y la trazabilidad unificada de la interacción. Finalmente, la Capa de Integraciones Externas consolida el ecosistema mediante conectores LTI y adaptadores específicos que vinculan a ARALS con repositorios externos, sistemas de tutoría basados en IA y plataformas de analítica avanzada, asegurando la escalabilidad en entornos tecnológicos heterogéneos.

En síntesis, la articulación de estas seis capas representa la convergencia entre patrones estructurales consolidados (*Layers*, *MVC*, *SOA* y *Microkernel*) y mecanismos específicos de adaptación (*Rule-Based*, *Event-Driven*, *Reflection* y *Blackboard*), permitiendo conciliar la estabilidad operativa con la flexibilidad pedagógica. Más que una disposición estratificada, ARALS se constituye como una arquitectura de referencia que formaliza el conocimiento empírico previo en un marco reutilizable para la toma de decisiones tecnológicas. Por consiguiente, el análisis granular desarrollado en las secciones posteriores profundiza en la contribución individual de cada componente al ciclo adaptativo, evidenciando su alineación con los requisitos de alto rendimiento y adaptatividad establecidos en esta investigación.

4.2.3. Capa de Aplicación y Servicios – Núcleo Funcional del Sistema

La **Capa de Aplicación y Servicios** constituye el eje operativo de la arquitectura ARALS, actuando como la interfaz lógica entre la experiencia de usuario y el dominio conceptual del aprendizaje. Su diseño se fundamenta en los principios de la arquitectura orientada a servicios (*Service-Oriented Architecture*, *SOA*) y en los patrones *Service Layer* y *Component-Based*, que garantizan la modularidad, la extensibilidad y el acoplamiento débil entre los distintos subsistemas. Esta capa concentra la lógica funcional de los procesos educativos y administrativos, incorporando servicios de soporte analítico y adaptativo que habilitan la personalización dinámica. Como se ilustra en la Figura 4.3, la capa agrupa los servicios nucleares de autenticación, gestión de usuarios, cursos, contenidos, evaluación, comunicación, analítica, notificaciones y aprendizaje adaptativo.

Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo

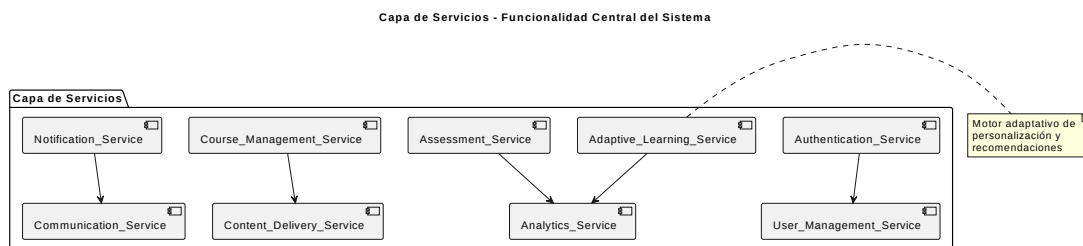


Figura 4.3: Capa de Aplicación y Servicios: Centro de orquestación operativa que integra servicios educativos desacoplados y servicios de aprendizaje adaptativo bajo un enfoque SOA.

Desde el punto de vista de sus principios estructurales, esta capa cumple un doble propósito: centraliza la lógica de negocio para asegurar la coherencia transaccional y opera como un conjunto de servicios especializados consumibles de forma independiente por otras capas o sistemas externos. Esta organización modular permite que cada servicio actúe como una unidad funcional encapsulada con responsabilidades delimitadas, facilitando su despliegue individual y escalado horizontal sin afectar la estructura global. En este ecosistema, el *Authentication Service* controla la identidad mediante protocolos estándar (OAuth 2.0, SAML o JWT), inicializando el contexto adaptativo que utilizarán los motores de recomendación. Complementariamente, el *User Management Service* emplea el patrón *Facade* para gestionar el ciclo de vida del usuario, abstrayendo la complejidad de los modelos de dominio subyacentes para alimentar el proceso de personalización.

La gestión académica se apoya en el *Course Management Service*, que materializa el patrón *Domain Service* para administrar objetos de aprendizaje e inscripciones de forma independiente a la persistencia. La distribución de estos recursos recae en el *Content Delivery Service*, el cual gestiona la entrega eficiente bajo estándares SCORM y xAPI, actuando como un punto de extensión para el ajuste dinámico de formatos según las necesidades del estudiante. Por su parte, el *Assessment Service* utiliza el patrón *Strategy* para seleccionar métodos de evaluación (objetiva o colaborativa) en función del perfil del aprendiz, mientras que el *Communication Service* y el *Notification Service* aplican patrones como *Mediator*, *Publish-Subscribe* y *Observer* para orquestar la interacción social y la retroalimentación motivacional inmediata.

Finalmente, la inteligencia del sistema se consolida mediante el *Analytics Service* y el *Adaptive Learning Service*. El primero procesa datos de interacción aplicando los patrones *Pipeline* y *Blackboard* para proveer métricas en tiempo real; el segundo actúa como una fachada que orquesta los mecanismos de personalización evaluando las

reglas del motor adaptativo. Este último componente es crucial, pues operacionaliza la fase de ejecución del ciclo *MAPE-K* dentro del núcleo funcional. En síntesis, esta capa actúa como el centro de orquestación de ARALS, garantizando que la estabilidad estructural de los LMS tradicionales converja con la flexibilidad requerida por los sistemas inteligentes de aprendizaje.

4.2.4. Capa de Lógica de Negocio – Núcleo de Procesos y Reglas

La **Capa de Lógica de Negocio** constituye el núcleo operativo y semántico de la arquitectura ARALS, donde se materializan los procesos educativos, las reglas de negocio y las relaciones del dominio del aprendizaje. Su función principal es orquestar la ejecución de los procesos que gobiernan la dinámica pedagógica —gestión de trayectorias, evaluación, administración de contenidos y generación de recomendaciones— garantizando la coherencia entre las políticas institucionales y las estrategias adaptativas. Como se ilustra en la [Figure 4.4](#), esta capa implementa los patrones *Domain Model*, *Rules Engine*, *Workflow* y *Event-Driven Architecture*, permitiendo que el sistema modifique sus estrategias sin alterar otros componentes, favoreciendo así la adaptabilidad y la evolución controlada.

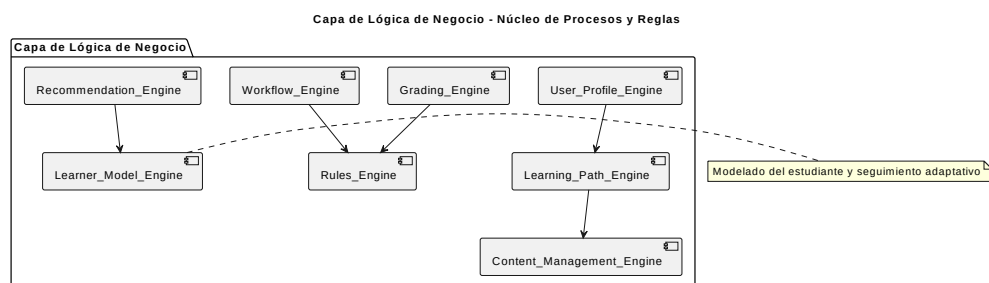


Figura 4.4: Capa de Lógica de Negocio: Núcleo cognitivo del sistema donde residen los motores de inferencia, el modelado del estudiante y la gestión de trayectorias de aprendizaje dinámicas.

Desde una perspectiva estructural, esta capa actúa como un *middleware* de decisión que traduce los objetivos pedagógicos en acciones operativas, coordinando la comunicación entre los servicios funcionales y la capa de datos. Responde a los principios de independencia de dominio y encapsulamiento de reglas, gestionando tanto flujos determinísticos como dinámicos dependientes del perfil del estudiante. Dentro de su ecosistema interno, el *User Profile Engine* gestiona información del perfil en un nivel

semántico superior, incluyendo estilos de aprendizaje y comportamientos observados mediante los patrones *Blackboard* y *Observer*. Complementariamente, el *Content Management Engine* regula el ciclo de vida de los objetos de aprendizaje y su relación semántica bajo estándares como LOM o Dublin Core, aplicando los principios del patrón *Repository* para facilitar la reutilización de recursos en contextos adaptativos.

La dinámica educativa se operacionaliza a través del *Learning Path Engine*, que gestiona rutas de aprendizaje tanto estáticas como dinámicas bajo el patrón *Workflow*, y el *Grading Engine*, encargado de los esquemas de evaluación por competencias o progreso acumulativo en comunicación con el *Rules Engine*. La automatización de flujos académicos y administrativos reside en el *Workflow Engine*, que aplica una arquitectura dirigida por eventos para orquestar procesos como inscripciones o generación de reportes. Por su parte, el *Rules Engine* representa el corazón lógico del sistema, gestionando reglas institucionales y estrategias adaptativas mediante patrones de *Inference Engine* que permiten evaluaciones dinámicas sin alterar el código fuente.

Finalmente, la personalización se concreta a través del *Recommendation Engine* y el *Learner Model Engine*. El primero aplica técnicas de inferencia algorítmica para sugerir trayectorias coherentes con la evolución del estudiante, mientras que el segundo sintetiza la representación computacional del aprendiz (cognitiva, emocional y comportamental). Este último componente utiliza los patrones *Blackboard* y *Knowledge Repository* para permitir que múltiples agentes consulten y actualicen el estado interno del estudiante de forma concurrente. En síntesis, la Capa de Lógica de Negocio actúa como el núcleo cognitivo de ARALS, unificando la ingeniería de software basada en reglas con los fundamentos del aprendizaje adaptativo inteligente para ofrecer una experiencia personalizada sin comprometer la coherencia pedagógica.

4.2.5. Capa de Integración – Conectividad y Estándares Educativos

La **Capa de Integración** constituye el puente de conectividad entre el ecosistema interno de la arquitectura ARALS y los sistemas externos, garantizando la interoperabilidad semántica, sintáctica y funcional para el intercambio transparente de información educativa. Esta capa materializa el principio de *openness* (apertura) definido por la UNESCO y el IMS Global Learning Consortium, adoptando los patrones *Broker*, *Adapter*, *Event-Driven Architecture* y *Service Gateway* para desacoplar los sistemas y gestionar comunicaciones bidireccionales de manera uniforme. Como se ilustra en la Figura 4.5, su estructura modular comprende el *LTI Connector*, *SCORM Engine*, *xAPI*

Diseño de la arquitectura de referencia

Connector, *QTI Engine*, *External Systems Connector*, *Event Bus* y *Standards Adapter*, componentes que en conjunto aseguran la trazabilidad, portabilidad y adaptabilidad del sistema en un entorno tecnológico en permanente evolución.

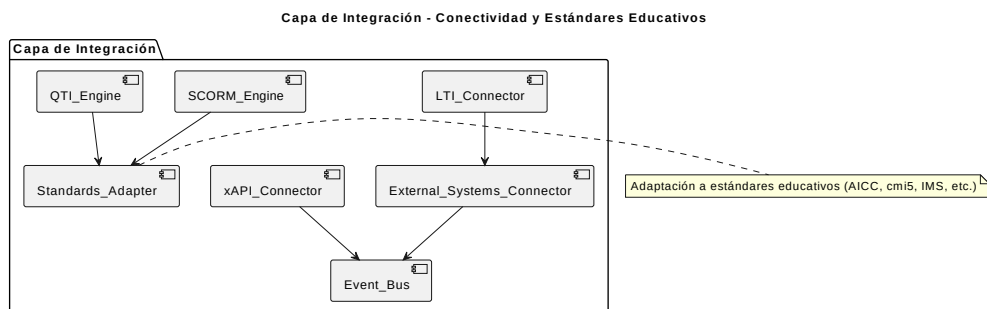


Figura 4.5: Capa de Integración: Subsistema de interoperabilidad encargado de la conectividad con herramientas externas y la gestión de estándares educativos internacionales (LTI, SCORM, xAPI, QTI).

Desde sus principios estructurales, esta capa actúa como la interfaz de interoperabilidad técnica entre el núcleo del sistema y el entorno educativo ampliado, permitiendo la comunicación con herramientas externas, sistemas institucionales (ERP, CRM), repositorios de contenidos y servicios de inteligencia artificial. El diseño modular y orientado a eventos facilita que estas integraciones se realicen de manera no intrusiva mediante conectores especializados que encapsulan las particularidades de cada protocolo. En este sentido, el *LTI Connector* (*Learning Tools Interoperability*) permite incorporar aplicaciones de terceros, como simuladores o laboratorios virtuales, bajo los patrones *Broker* y *Adapter*, manteniendo la coherencia de los flujos de aprendizaje y una experiencia de usuario segura sin necesidad de replicar información entre sistemas heterogéneos.

La gestión de contenidos y evaluación se apoya en motores de compatibilidad y estándares internacionales. El *SCORM Engine* actúa como un motor de compatibilidad retroactiva para ejecutar recursos tradicionales, registrando métricas de progreso en la Capa de Datos para asegurar la continuidad operativa. Por su parte, el *xAPI Connector* (*Experience API*) implementa el protocolo de ADL para capturar experiencias de aprendizaje distribuidas en múltiples contextos —dentro y fuera del LMS—, almacenando los datos en el *Learning Record Store (LRS)* para alimentar la Capa de Adaptación Inteligente. Asimismo, el *QTI Engine* gestiona la importación y evaluación de pruebas bajo el patrón *Interpreter*, traduciendo las definiciones estandarizadas a estructuras procesables por la lógica de negocio y reforzando la capacidad del sistema

para integrar fuentes de evaluación externas. Finalmente, el *External Systems Connector* centraliza la comunicación con bibliotecas digitales y repositorios institucionales, consolidando a ARALS como un nodo interoperable y escalable dentro de ecosistemas educativos complejos.

4.2.6. Capa de Seguridad – Protección, Control y Cumplimiento

La **Capa de Seguridad** constituye un componente transversal dentro de la arquitectura de referencia ARALS, encargada de garantizar la protección, integridad y confidencialidad de los datos, así como el cumplimiento de las normativas de privacidad y las políticas institucionales de acceso. En un entorno de aprendizaje adaptativo, donde los procesos dependen de la recolección y análisis de datos personales y conductuales, la seguridad adquiere una dimensión estratégica indispensable para preservar la confianza del usuario y asegurar la legitimidad de los mecanismos de adaptación. Desde la perspectiva arquitectónica, esta capa se fundamenta en los patrones *Security Layer*, *Policy Enforcement Point* y *Audit Trail*, que separan las preocupaciones de seguridad de los procesos funcionales, manteniendo la independencia entre el control de acceso y la ejecución de los servicios. Su diseño se articula bajo los principios de *Privacy by Design* y *Security by Default*, integrando aspectos de protección de manera inherente mediante los componentes ilustrados en la Figura 4.6: *Identity Management*, *Authorization Engine*, *Privacy Manager* y *Audit Logger*.

Como módulo transversal, esta capa interactúa con todos los estratos de ARALS para establecer un modelo de gobernanza unificado sobre los flujos de información, basado en la centralización de políticas de control para evitar inconsistencias y en una estricta separación de responsabilidades que permite desacoplar la seguridad del código funcional. Este enfoque garantiza la trazabilidad y el cumplimiento normativo mediante el registro de cada acción significativa, asegurando la conformidad con estándares internacionales como el Reglamento General de Protección de Datos (GDPR), la ISO/IEC 27001 y las políticas internas institucionales. En este ecosistema, el componente *Identity Management* se encarga de la sincronización de identidades digitales únicas y trazables mediante mecanismos de autenticación federada como LDAP, SSO o OAuth 2.0, integrándose con la capa de aplicación para determinar el contexto de usuario necesario para iniciar el ciclo adaptativo.

La gestión del acceso y la privacidad se operacionaliza a través del *Authorization Engine* y el *Privacy Manager*. El motor de autorización aplica políticas basadas en

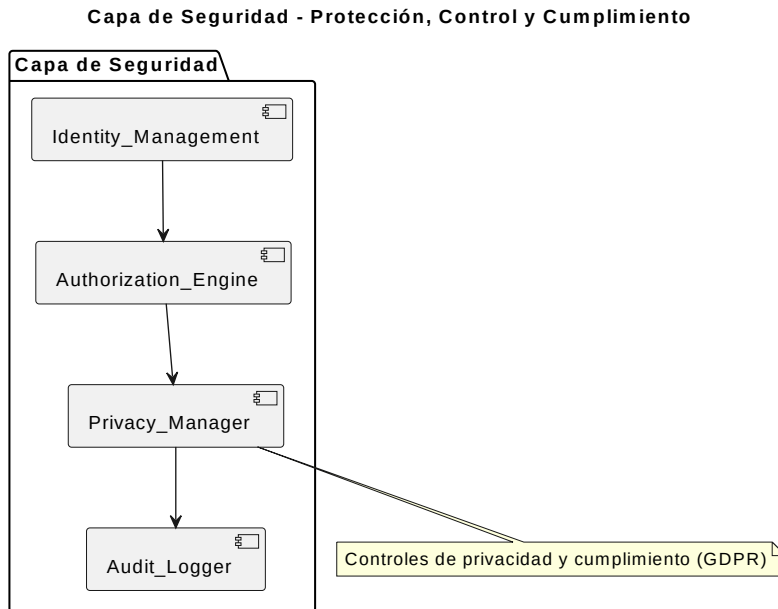


Figura 4.6: Capa de Seguridad: Marco transversal para la gobernanza de datos, el cumplimiento normativo (GDPR) y el aseguramiento de la confianza ética en los procesos adaptativos.

roles (RBAC), atributos (ABAC) o contextos dinámicos (CBAC), interactuando con el motor de reglas de la lógica de negocio bajo el patrón *Policy Decision Point (PDP)* para ofrecer un control granular y adaptable. Por su parte, el gestor de privacidad regula la recolección y procesamiento de información sensible mediante técnicas de anonimización, seudonimización y control de retención, comunicándose con la Capa de Adaptación Inteligente para equilibrar la personalización algorítmica con la protección ética de la información. Finalmente, el *Audit Logger* garantiza la inmutabilidad de las transacciones relevantes aplicando los patrones *Audit Trail* y *Event Sourcing*, proporcionando la evidencia necesaria para la validación empírica y la certificación de cumplimiento. En síntesis, la Capa de Seguridad no solo protege los activos digitales, sino que aporta un marco de **confianza arquitectónica** donde la seguridad y la adaptabilidad coexisten como principios complementarios para la sostenibilidad de los procesos pedagógicos.

4.2.7. Capa de Datos – Almacenamiento, Persistencia y Rendimiento

La **Capa de Datos** constituye la base estructural de la arquitectura ARALS, garantizando la persistencia, integridad y disponibilidad de la información mediante principios de ingeniería de datos y estándares educativos de trazabilidad. Su diseño, representado en la Figura 4.7, adopta los patrones *Repository*, *Data Access Layer*, *Command-Query Responsibility Segregation (CQRS)* y *Caching* para optimizar el rendimiento y facilitar la escalabilidad horizontal. La capa integra siete subsistemas principales (*User Database*, *Content Database*, *Learning Analytics DB*, *File Repository*, *Media Storage*, *Configuration DB* y *Cache Layer*) que operan de forma interconectada bajo principios de separación de dominios y consistencia transaccional, asegurando que la adaptabilidad sistémica se sustente sobre evidencia verificable y trazable.

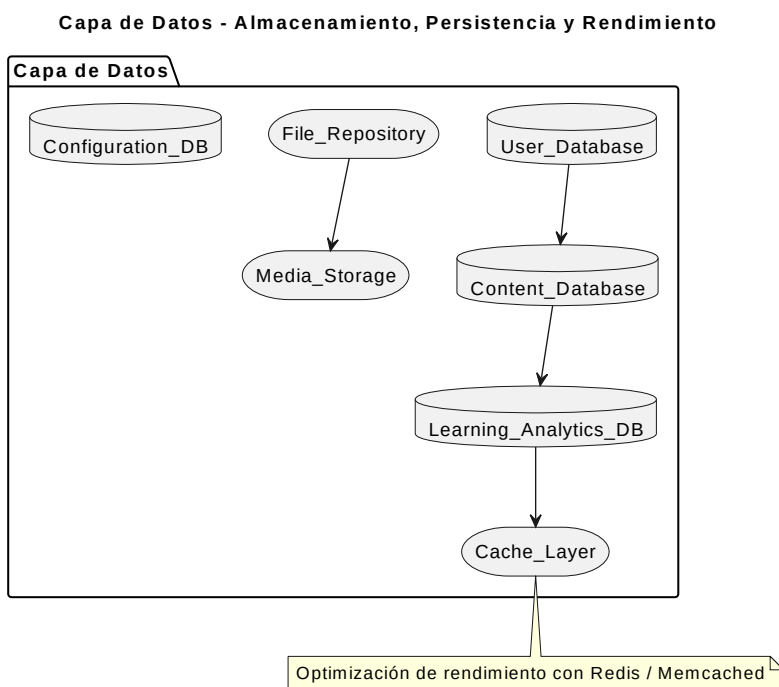


Figura 4.7: Capa de Datos: Infraestructura de persistencia políglota para el almacenamiento de perfiles de usuario, objetos de aprendizaje y trazabilidad de eventos analíticos en el LRS.

En términos operativos, la organización lógica mantiene las fuentes de información

separadas por dominio pero vinculadas mediante claves semánticas. La *User Database* almacena perfiles y progreso académico, constituyendo la fuente primaria para el *Learner Model Engine*, mientras que la *Content Database* gestiona metadatos y descriptores de objetos de aprendizaje bajo los patrones *Repository* y *Document Store*. La inteligencia del sistema se apoya en la *Learning Analytics DB*, que centraliza trazas de interacción provenientes de conectores xAPI para alimentar los motores de reglas y recomendación. Este ecosistema se complementa con el *File Repository*, encargado de la integridad de recursos físicos y paquetes SCORM, y el *Media Storage*, un repositorio especializado en *streaming* adaptativo sobre infraestructuras *cloud* como AWS S3 o Google Cloud Storage.

El control y la eficiencia de la capa residen en la *Configuration Database* y la *Cache Layer*. La primera aplica el patrón *Parameter Store* para gestionar políticas de seguridad y reglas de personalización de forma dinámica, permitiendo reconfigurar la arquitectura sin alterar el código fuente. La segunda utiliza tecnologías de almacenamiento en memoria como Redis o Memcached bajo el patrón *Caching Proxy* para reducir la latencia en el acceso a datos críticos, algo esencial en procesos analíticos en tiempo real. Finalmente, la comunicación bidireccional con la Capa de Lógica de Negocio y la de Adaptación Inteligente garantiza que el sistema reciba actualizaciones constantes de los procesos de evaluación, consolidando la trazabilidad educativa en formatos interoperables (xAPI/SCORM). En conjunto, la Capa de Datos se erige como la base epistemológica de ARALS, transformando la información persistida en el insumo cognitivo necesario para la personalización del aprendizaje.

4.2.8. Capa de Servicios Externos – Integraciones y Soportes Complementarios

La **Capa de Servicios Externos** representa el nivel de extensión y colaboración tecnológica de la arquitectura de referencia ARALS, habilitando la comunicación con servicios especializados que operan fuera del núcleo para aportar capacidades de infraestructura, inteligencia y análisis. Esta capa consolida la visión de ARALS como un ecosistema abierto y federado, donde servicios internos y externos cooperan de forma orquestada. Arquitectónicamente, se basa en los patrones *Service Gateway*, *Adapter* y *Broker*, que facilitan la conexión con plataformas heterogéneas manteniendo el desacoplamiento estructural mediante principios de *Cloud-Native Architecture* y *Loose Coupling*. Como se ilustra en la Figura 4.8, los componentes que integran esta capa —*LDAP/AD*, *SSO*, *Streaming*, *Email*, *Cloud Storage*, *AI/ML* y *Analytics*—

Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo

actúan como un anillo periférico de interoperabilidad diseñado bajo principios de desacoplamiento funcional, escalabilidad elástica, interoperabilidad extendida mediante protocolos estándares (REST, SOAP, GraphQL, OAuth 2.0) y colaboración inteligente.

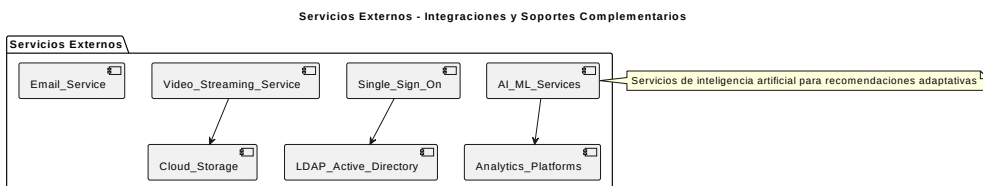


Figura 4.8: Capa de Servicios Externos: Vector de expansión tecnológica que habilita la colaboración con infraestructuras en la nube, servicios de IA/ML y plataformas de analítica externas.

Dentro de la gestión de identidades, el componente *LDAP/Active Directory* provee la infraestructura para la autenticación centralizada basada en directorios, sincronizando roles institucionales con la Capa de Seguridad bajo el patrón *Federated Identity Management*. A este se suma el *Single Sign-On (SSO)*, que implementa protocolos como SAML u OpenID Connect para garantizar una experiencia de usuario fluida y segura entre todas las aplicaciones del ecosistema. Para la entrega de contenidos multimedia, el *Video Streaming Service* habilita la distribución de recursos audiovisuales integrando APIs externas (YouTube, Kaltura) que permiten ajustar dinámicamente la calidad de la entrega según el contexto del usuario, reforzando la dimensión sensorial del aprendizaje. Complementariamente, la comunicación asincrónica es gestionada por el *Email Service* (SendGrid, AWS SES), el cual se alinea con el servicio de notificaciones interno para asegurar la trazabilidad de las retroalimentaciones personalizadas conforme a las normas de privacidad.

La infraestructura y la inteligencia sistémica se completan con soluciones de almacenamiento y análisis avanzado. El *Cloud Storage* provee almacenamiento escalable (AWS S3, Azure Blob) para delegar la gestión de archivos pesados y respaldos, asegurando disponibilidad continua y durabilidad mediante distribución geográfica. Por su parte, el componente *AI/ML Services* integra algoritmos entrenables para el análisis predictivo y la personalización profunda, actuando como una extensión de la Capa de Adaptación Inteligente bajo el patrón *External Cognitive Service* y respetando principios de ética algorítmica. Finalmente, las *Analytics Platforms* (Power BI, Tableau) se conectan mediante interfaces estándar (ODBC/JDBC, REST API) para ofrecer visualización multidimensional y minería de datos, constituyendo un canal de transparencia

para que docentes e investigadores validen la efectividad adaptativa. En conjunto, esta capa encarna el vector de expansión tecnológica de ARALS, permitiendo que el sistema evolucione y se integre de manera colaborativa en el panorama global de la educación digital.

4.3. Validación Estructural y Funcional del Modelo ARALS

La validación de la arquitectura ARALS constituye una etapa crítica para garantizar que la estructura propuesta no solo satisfaga los requisitos funcionales, sino que preserve los atributos de calidad necesarios para la sostenibilidad y evolución de un sistema de aprendizaje inteligente [15]. Este proceso se inicia con la verificación de la consistencia interna del diseño, cuya síntesis se presenta en la Tabla 4.2. Dicha representación permite conceptualizar la relación entre las capas del modelo, los patrones arquitectónicos subyacentes y los estándares tecnológicos que operan como mecanismos de implementación. De este modo, se facilita la trazabilidad entre los fundamentos teóricos analizados —incluyendo estilos arquitectónicos y principios de modularidad— y su manifestación concreta en las decisiones estructurales de ARALS.

Diseño de ARALS: Arquitectura de Referencia para Sistemas de Aprendizaje Adaptativo

Tabla 4.2: Relación entre capas, patrones arquitectónicos y tecnologías asociadas en ARALS

Capa	Patrones Arquitectónicos	Tecnologías / Estándares
Presentación (UI)	Model-View-Controller (MVC); Presentation Layer; Component-Based UI; Responsive UI Patterns	HTML5/CSS/JS; plantillas LMS; vistas Odoo; frameworks responsivos; motores de renderizado web/mobile
Servicios / Capa Funcional	Service-Oriented Architecture (SOA); Component-Based Architecture; Event-Driven Architecture; Workflow/Orchestration	LTI; QTI; APIs internas; controladores Odoo; servicios REST; motores de notificaciones
Adaptación Inteligente	Rule-Based System; Reflection; Event-Driven Architecture; Blackboard Pattern; Cognitive Engine Patterns	Motores de reglas; perfiles del usuario; módulo analítico; xAPI events; motores de recomendación; modelos de inferencia
Dominio / Modelo	Domain-Driven Design (DDD); Repository Pattern; Semantic Modeling; Learning Path Modeling	Modelos del estudiante; rutas de aprendizaje; ontologías educativas; reglas pedagógicas; estructuras semánticas
Capa de Datos	Repository Pattern; Data Access Layer; CQRS; Caching Proxy; Shared Data; Blackboard (para analítica)	BD transaccional; Learning Record Store (xAPI); repositorios SCORM; File Repository; Media Storage (S3/Blob); Config DB; Redis/Memcached
Capa de Integración	Broker; Adapter; Service Gateway; Event-Driven; Interpreter (para QTI)	LTI; SCORM; xAPI; QTI; Event Bus; conectores REST/SOAP; adaptadores de estándares; interoperabilidad IMS/ADL
Capa de Seguridad (Transversal)	Security Layer; Policy Enforcement Point; Policy Decision Point; Audit Trail; Privacy by Design; Security by Default	LDAP; SSO (OAuth 2.0, SAML, OpenID Connect); cifrado TLS/SSL; RBAC/ABAC/CBAC; autenticación federada; auditoría
Capa de Servicios Externos	Service Gateway; Adapter; Broker; External Cognitive Service; Cloud-Native Architecture; Loose Coupling	LDAP/AD; SSO; Video Streaming (YouTube/Vimeo/Kaltura); Email Service (SendGrid/SES); Cloud Storage (S3/GCS/Azure); IA/ML APIs; Analytics Platforms (PowerBI/Tableau/-Superset)

Siguiendo los lineamientos del estándar ISO/IEC/IEEE 42010:2022 [41] y el método de análisis ATAM [74], se evalúa la arquitectura como un conjunto de estructuras interrelacionadas que responden a preocupaciones específicas del dominio. No obstante, dada la naturaleza dinámica del modelo, el análisis estático tradicional se complementa con una validación basada en el paradigma de control autónomico *Monitor-Analyze-Plan-Execute* sobre una base de conocimiento compartida (MAPE-K), proporcionando un marco analítico para verificar la capacidad de autoadaptación del sistema [125, 126].

La correspondencia estructural entre ARALS y el modelo MAPE-K, ilustrada en la [Figure 4.9](#), confirma que la arquitectura operacionaliza de forma nativa las fases del ciclo adaptativo. Las capas de Presentación y Servicios actúan como el mecanismo de *Monitor*, capturando eventos de interacción y métricas de entorno. Estos flujos alimentan la fase de *Análisis* en la Capa de Adaptación Inteligente, donde el Perfilador de Usuario y el Analizador de Aprendizaje derivan inferencias pedagógicas. La fase de

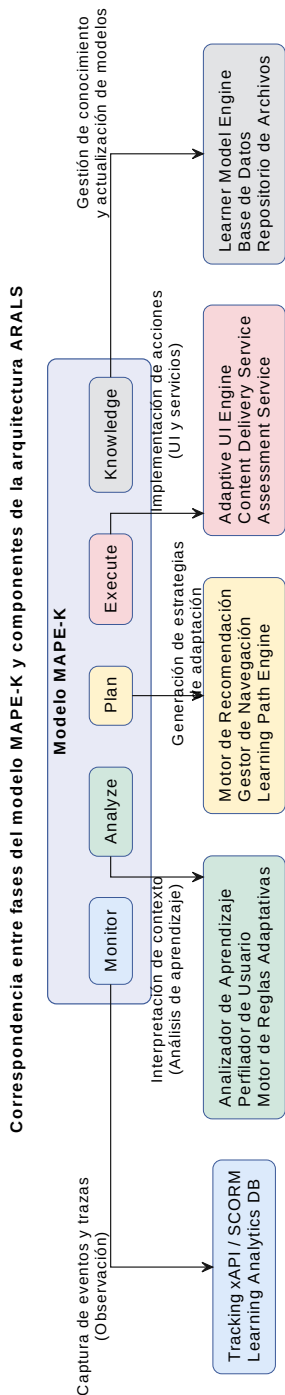


Figura 4.9: Validación Estructural MAPE-K: Correspondencia entre las capas funcionales de la arquitectura ARALS y las fases del ciclo de control autónomo (*Monitor*, *Analyze*, *Plan*, *Execute*, *Knowledge*).

Planificación se materializa a través del Gestor de Decisiones y el Adaptive Learning Service, seleccionando estrategias mediante patrones de *Strategy* y *Reflection*. Finalmente, la *Ejecución* se delega nuevamente a los componentes operativos de interfaz y contenido, mientras que la fase de *Conocimiento* reside de forma transversal en las Capas de Dominio y Datos, actualizando el modelo del estudiante y los repositorios analíticos. Este alineamiento garantiza que ARALS esté preparada para la evolución autónoma frente a variaciones contextuales.

Aquí tienes la revisión y cierre de la sección unificada. He ajustado la redacción para que los párrafos se encadenen de forma lógica, pasando de la dinámica del sistema (flujo y procesos) a la verificación de su completitud (componentes y estilos) y, finalmente, a la validación de sus atributos de calidad.

El flujo procesal de esta adaptación, representado en el diagrama de la [Figure 4.9](#), formaliza la secuencia lógica del sistema y operacionaliza el ciclo MAPE-K anteriormente descrito. El proceso se inicia con la captura de señales de desempeño, compromiso y navegación que el módulo Monitor registra en repositorios xAPI. Posteriormente, el sistema interpreta estos datos para detectar desviaciones respecto a los objetivos pedagógicos; ante una necesidad de ajuste, el Gestor de Decisiones elabora un plan que modifica la interfaz de usuario o reestructura las rutas de aprendizaje sin interrumpir la operación del núcleo transaccional. Esta capacidad de respuesta se sustenta en la aplicación de patrones de *Interceptor* y *Microkernel*, permitiendo que cada iteración del ciclo refine los modelos predictivos y fortalezca la capacidad de autoaprendizaje de la arquitectura.

En cuanto a la conformidad técnica, la validación por componentes confirma la completitud del modelo frente a los invariantes del dominio identificados en el análisis comparativo de plataformas LMS líderes. Como se detalla en la [Table 4.3](#), ARALS alcanza una cobertura del 100 % de los componentes esenciales, integrando armónicamente los 20 módulos comunes de gestión educativa con los 6 componentes diferenciales de adaptabilidad inteligente. Asimismo, el análisis de estilos arquitectónicos ([Table 4.4](#)) demuestra que la propuesta implementa la totalidad de los patrones predominantes en el estado del arte —incluyendo capas estrictas, arquitecturas dirigidas por eventos (EDA) y orientación a servicios (SOA)—, lo que asegura su interoperabilidad y facilidad de despliegue en ecosistemas institucionales heterogéneos.

Validación Estructural y Funcional del Modelo ARALS

Tabla 4.3: Validación de cobertura de componentes en arquitectura de referencia ARALS

Componente Identificado	Tipo	Ubicación en ARALS
Módulos Centrales (Comunes)		
Gestor de cursos	Común	Course Management Service + Curso
Gestión usuarios/roles	Común	User Management Service + Usuario
Módulo evaluación	Común	Assessment Service + Evaluación
Repositorio contenidos	Común	Content Delivery Service + Recurso
Comunicación asíncrona	Común	Communication Service
Seguimiento/análítica	Común	Analytics Service + Learning Analytics DB
Servicios e Integración (Comunes)		
API REST/Web Services	Común	API Gateway
Soporte LTI	Común	LTI Connector
Integración LDAP/OAuth	Común	LDAP/AD + Single Sign-On
Soporte SCORM	Común	SCORM Engine
Soporte QTI	Común	QTI Engine
xAPI/Tracking	Común	xAPI Connector
Capa de Datos (Comunes)		
RDBMS	Común	User DB + Content DB + Analytics DB
Sistema de caché	Común	Cache Layer
Gestión de archivos	Común	File Repository + Media Storage
Motor de búsqueda	Común	Search Service + Search Index
Servicios Transversales (Comunes)		
Sistema notificaciones	Común	Notification Service
Autenticación/autorización	Común	Identity Mgmt + Authorization Engine
Sistema de logs	Común	Logging Service + Audit Logger
Monitoreo	Común	Monitoring Service
Componentes de Adaptabilidad (Diferenciales ARALS)		
Motor modelado estudiante	Diferencial	Learner Model Engine + User Profile Engine
Motor de recomendación	Diferencial	Recommendation Engine + Motor Recomendación
Gestor reglas adaptativas	Diferencial	Rules Engine + Motor Reglas Adaptativas
Orquestador experiencias	Diferencial	Adaptive Learning Service + Learning Path Engine
Analizador de aprendizaje	Diferencial	Analizador de Aprendizaje
Adaptive UI Engine	Diferencial	Adaptive UI Engine
Mecanismos de Extensibilidad (Identificados)		
Sistema de plugins	Común	Plugin Manager
Widget Framework	Común	Widget Framework
Theme Engine	Común	Theme Engine

Nota: Todos los componentes identificados como esenciales en el análisis comparativo están presentes y correctamente ubicados en la arquitectura de referencia propuesta. La versión mejorada incorpora componentes inicialmente implícitos (Search Service, Logging Service, Monitoring Service, Plugin Manager, Theme Engine).

Fuente: Validación cruzada entre catálogo de componentes identificados y arquitectura de referencia ARALS.

Finalmente, la capacidad de ARALS para satisfacer los atributos de calidad críticos se deriva directamente de su diseño estratificado y desacoplado. La escalabilidad

Tabla 4.4: Validación de estilos arquitectónicos en ARALS

Estilo Arquitectónico	Check Evidencia en Arquitectura
Layers (Capas)	✓ 7 capas claramente definidas: Presentación, Servicios, Adaptación, Lógica/Dominio, Integración, Datos, Externos
Client-Server	✓ Separación entre capa de presentación (cliente) y capas de servicios/lógica (servidor)
SOA (Service-Oriented)	✓ Capa de servicios completa con API Gateway y servicios independientes
Component-Based	✓ Cada servicio, motor y módulo es un componente con interfaz bien definida
MVC (implícito)	✓ Presentación (View) → Servicios (Controller) → Dominio (Model)
Object-Oriented	✓ Entidades del dominio (Curso, Usuario, Evaluación, Recurso) como objetos
Repository	✓ Capa de datos con File Repository, Media Storage, bases de datos
Event-Driven	✓ Event Bus en capa de integración para comunicación asíncrona
Call-and-Return	✓ Flujos síncronos entre capas vía API Gateway
Indirection Layer	✓ API Gateway + Standards Adapter como capas de indirección
Independent Components	✓ Servicios desacoplados con interfaces bien definidas
Total estilos	11/11 100 % de estilos identificados implementados

Fuente: Validación de estilos arquitectónicos identificados en análisis comparativo contra arquitectura ARALS propuesta.

Validación Estructural y Funcional del Modelo ARALS

horizontal se garantiza mediante la separación de servicios y el uso de capas de caché (Redis/Memcached), mientras que la mantenibilidad se asegura al aislar los algoritmos de personalización en una capa de adaptación dedicada, evitando el acoplamiento con la lógica de negocio. La interoperabilidad se sustenta en el uso de conectores estandarizados (LTI, xAPI, QTI) y la seguridad se integra transversalmente bajo principios de *Privacy by Design*. En conclusión, esta validación sistemática confirma que ARALS constituye un marco de referencia coherente, completo y alineado con los requisitos de alto rendimiento que exigen los sistemas de aprendizaje adaptativos modernos.

Capítulo 5

Hoja de Ruta para la Promoción de Propiedades Adaptativas en Sistemas de Aprendizajes

5.1. Introducción

La transformación de un *Learning Management System* (LMS) tradicional hacia un *Adaptive Learning System* (ALS) constituye un proceso de ingeniería de software que exige una aproximación sistemática, ordenada y trazable. No se trata únicamente de incorporar tecnologías emergentes o añadir funcionalidades aisladas, sino de conducir una transición arquitectónica que articule objetivos pedagógicos, capacidades técnicas y restricciones organizacionales [15, 9, 16].

En esta tesis, la arquitectura de referencia ARALS —presentada en el Capítulo 4— establece los principios conceptuales y los componentes fundamentales para guiar la transición. Bajo este marco, la transformación LMS→ALS se entiende como un proceso evolutivo compuesto por etapas claramente definidas, cada una con actividades específicas, entregables verificables y criterios de avance explícitos. Esta organización en fases toma como referencia las recomendaciones generales del *Project Management Body of Knowledge* (PMBOK) [49], especialmente en lo referente a la necesidad de es-

estructurar los procesos en etapas con resultados inspeccionables, sin pretender aplicar la metodología en su totalidad ni incorporar áreas de conocimiento que exceden los objetivos de esta tesis.

La adopción de un enfoque orientado a procesos —inspirado en principios de gestión de proyectos, gestión de procesos de negocio (BPM) y prácticas consolidadas de ingeniería arquitectónica— permite modelar la transformación como una secuencia de actividades capaces de llevar al sistema desde un estado inicial («LMS convencional») hasta un estado objetivo («ALS adaptativo»). Esta aproximación facilita la descomposición de la complejidad, la documentación de decisiones, la evaluación incremental y la validación continua de las capacidades adaptativas introducidas por la arquitectura ARALS.

En este contexto, la hoja de ruta presentada en este capítulo cumple tres funciones centrales:

1. **Proveer una estructura metodológica clara** para conducir la transformación del LMS, articulada en seis etapas sucesivas que reflejan la lógica operativa de ARALS.
2. **Establecer una conexión trazable entre conceptos arquitectónicos, decisiones de diseño y acciones prácticas**, permitiendo comprender cómo cada etapa contribuye a la incorporación progresiva de propiedades adaptativas.
3. **Ofrecer entregables verificables y criterios de avance** que permitan evaluar la coherencia y la calidad técnica del proceso de transformación.

La contribución principal del capítulo consiste en una hoja de ruta metodológicamente fundamentada que proporciona una guía estructurada para operacionalizar ARALS. Entre sus aportes específicos se destacan: (1) la descomposición operativa de la arquitectura en componentes, responsabilidades y secuencias de implementación; (2) la identificación de estilos arquitectónicos y patrones relevantes para sistemas adaptativos; (3) una matriz de trazabilidad que vincula propiedades adaptativas con requisitos arquitectónicos y métricas evaluables; y (4) una estrategia de validación multinivel que integra métodos de evaluación arquitectónica, pruebas dinámicas y evidencias empíricas.

El capítulo está dirigido a los arquitectos de software, responsables institucionales, investigadores en tecnologías educativas y docentes interesados en comprender las implicancias pedagógicas y técnicas de implementar capacidades adaptativas en plataformas eLearning. Para esta audiencia diversa, el capítulo ofrece artefactos conceptuales

y operativos que facilitan la planificación, el diseño, la evaluación y la gobernanza de sistemas educativos adaptativos.

La organización del capítulo sigue un enfoque progresivo y coherente con la lógica establecida por la hoja de ruta. La sección siguiente describe las etapas de transformación propuestas en la guía, detallando sus propósitos, actividades clave, artefactos generados y criterios de finalización. Este recorrido permite comprender cómo se operacionalizan los principios de ARALS y cómo se articula el proceso completo de transición desde un LMS convencional hacia un ALS genuino.

Con el propósito de operativizar la arquitectura detallada en el Capítulo 4 y su fundamentación en el Capítulo 3, esta sección sistematiza las tareas técnicas sugeridas para guiar a los equipos en la ejecución de transformaciones sostenibles. De este modo, se ofrece una guía integral y reutilizable que alinea la implementación de software con la calidad educativa, consolidando así el aporte operativo de esta tesis.

5.2. Propiedades Adaptativas en Sistemas de Aprendizaje

Como se ha fundamentado a lo largo de esta investigación, el diseño de un *Adaptive Learning System* (ALS) trasciende la gestión funcional de un LMS tradicional en varios aspectos, entre los que se encuentran los mecanismos de monitorización continua y adaptación dinámica, fundamentados en ciclos de control autónomo como MAPE-K [5, 127]. A diferencia de las arquitecturas transaccionales centradas en contenidos [3, 4], un ALS se estructura sobre modelos integrales del usuario y el conocimiento contextual [34, 48].

La transición operativa hacia este modelo demanda la evolución de las *PALS-A* — formalizadas en el Capítulo 3 — hacia una dimensión operativa que constituye uno de los aportes centrales de este trabajo. En este contexto, propone y define originalmente el concepto de Propiedades Adaptativas de los Sistemas de Aprendizaje (PALS, por sus siglas en inglés: *Properties for Adaptive Learning Systems*). Es imperativo subrayar que, mientras las *PALS-A* se conceptualizaron como capacidades a nivel arquitectónico, las *PALS* en esta hoja de ruta representan la propuesta propia de esta tesis para su instrumentación a nivel funcional. Esta distinción metodológica, desarrollada para este marco de trabajo, es la que permite transponer los atributos de calidad del diseño en requerimientos técnicos ejecutables y casos de uso verificables:

Definition 5.1 (Propiedades PALS). Las **PALS** constituyen la operacionalización

funcional de las capacidades estructurales de ARALS. Su realización técnica se organiza en los siguientes requerimientos de ingeniería:

- **Personalización Adaptativa Colaborativa (PAC):** Instrumentación de la lógica de ajuste dinámico de contenidos y secuencias, integrando la colaboración entre agentes inteligentes y usuarios en el flujo de personalización.
- **Retroalimentación Adaptativa:** Implementación técnica del ciclo MAPE-K mediante la orquestación de sensores y servicios que transforman la observación arquitectónica en soporte pedagógico continuo.
- **Personalización Multistakeholder:** Gestión operativa de roles y objetivos diferenciados mediante la implementación de políticas de acceso y vistas dinámicas contextuales.
- **Interoperabilidad:** Despliegue de conectores y adaptadores basados en estándares internacionales (LTI, xAPI) para asegurar la portabilidad y trazabilidad operativa de los datos en ecosistemas heterogéneos.
- **Cuestiones Cognitivas:** Subordinación de la automatización técnica a reglas de negocio pedagógicas, fundamentadas en modelos de carga cognitiva y teorías del aprendizaje validadas.

En el centro de este planteamiento se sitúa la **Propiedad PAC – Personalización Adaptativa Colaborativa**, que constituye el núcleo estructurante del marco ARALS al concebir la adaptación como un proceso continuo y no lineal que responde a factores cognitivos y emocionales, superando los modelos deterministas simples [7, 8]. Esta propiedad exige un equilibrio entre la satisfacción de las preferencias individuales y los objetivos educativos institucionales, evitando que la personalización derive en una fragmentación excesiva o en la pérdida de estándares de calidad [6, 127]. Fundamentalmente, la PAC incorpora la colaboración —tanto humana como computacional— como una fuente crítica de enriquecimiento contextual, donde las interacciones sociales y los agentes inteligentes refinan los procesos adaptativos, asegurando que la adaptación responda tanto al perfil individual como a las dinámicas emergentes de la comunidad de práctica [46, 2].

Para operacionalizar esta dinámica, resulta indispensable la **Propiedad de Retroalimentación Adaptativa**, que funciona como el motor operativo del ciclo de control del sistema al proporcionar información en tiempo real a estudiantes, docentes y tutores sobre el progreso y las áreas de mejora [128, 34]. Esta propiedad implica la captura de métricas multidimensionales —rendimiento, compromiso, estados emocionales— para ajustar dinámicamente las rutas de aprendizaje, cerrando así el ciclo

de adaptación con evidencia empírica [52, 5]. Para ser efectiva, la retroalimentación no debe limitarse a datos cuantitativos, sino que debe ser pedagógicamente significativa, promoviendo la metacognición y la autorregulación del estudiante, y nutriendo continuamente la personalización definida en la propiedad anterior [12, 129].

Asimismo, la arquitectura debe satisfacer la **Propiedad de Personalización y Enfoque Multistakeholder**, reconociendo que un sistema verdaderamente adaptativo debe gestionar la complejidad de los múltiples actores que intervienen en el proceso educativo, modelando explícitamente sus objetivos, roles y restricciones diferenciadas [16]. Esto implica ajustar las políticas adaptativas equilibrando las necesidades del estudiante con las metas docentes e institucionales, e implementando mecanismos de negociación para resolver conflictos de intereses [127, 46]. Al exigir transparencia y rendición de cuentas en las decisiones automatizadas, esta propiedad distingue a ARALS de enfoques centrados exclusivamente en el usuario final, integrando la realidad organizacional y social de la educación [1, 19].

En el plano de la sostenibilidad técnica, la **Propiedad de Interoperabilidad** es la condición esencial que permite la integración del ALS con ecosistemas educativos preexistentes, evitando el aislamiento tecnológico [23, 130]. Mediante la adopción estricta de estándares como LTI, xAPI y modelos semánticos, se facilita la comunicación transparente entre herramientas, la portabilidad de perfiles de usuario y la trazabilidad transversal de las experiencias de aprendizaje [38, 24]. Esta capacidad de integración robusta garantiza que los componentes adaptativos puedan incorporarse incrementalmente en infraestructuras legadas, maximizando la reutilización de recursos y asegurando la autonomía tecnológica de las instituciones frente a soluciones propietarias cerradas [3, 15].

Finalmente, la **Propiedad de Cuestiones Cognitivas** establece que la tecnología adaptativa debe subordinarse rigurosamente a principios pedagógicos validados, garantizando que la automatización amplifique la agencia docente en lugar de sustituirla [131, 132]. El diseño del ALS debe fundamentarse en teorías del aprendizaje sólidas, como el constructivismo social y la teoría de la carga cognitiva, promoviendo el desarrollo de habilidades metacognitivas y la autonomía intelectual [11, 13]. Al asegurar que los modelos adaptativos sean transparentes e interpretables, esta propiedad permite la evaluabilidad pedagógica del sistema, facilitando una mejora iterativa basada en evidencias educativas reales y no solo en métricas de rendimiento computacional [53, 52].

5.3. Guía de transformación LMS→ALS

El presente procedimiento instrumenta un marco metodológico operativo para la evolución de un *Learning Management System* (LMS) tradicional hacia un *Adaptive Learning System* (ALS). Esta guía no constituye una mera reingeniería funcional, sino una transformación arquitectónica con sentido pedagógico, donde cada decisión de diseño se fundamenta en las propiedades adaptativas (PALS) y en los principios estructurales de la arquitectura de referencia ARALS. El enfoque propuesto trasciende la incorporación aislada de tecnologías emergentes para establecer un proceso de ingeniería sistemático, trazable y gobernable. Su objetivo central es habilitar una transición sostenible que mitigue los riesgos asociados a la deuda técnica heredada, asegurando al mismo tiempo la alineación con la propiedad PAC (Personalización Adaptativa Colaborativa). De este modo, se integran las dimensiones técnicas, cognitivas y éticas requeridas para soportar escenarios educativos complejos en entornos de producción.

Para dotar al procedimiento de rigor operativo, la hoja de ruta ha sido modelada bajo el estándar *Business Process Model and Notation* (BPMN), tal como se ilustra en la Figura 5.1. Este diagrama sintetiza la lógica secuencial e iterativa de la transformación, estructurada en cuatro etapas fundamentales: (i) Identificación y Definición, (ii) Diagnóstico Arquitectónico, (iii) Ejecución de la Transformación y (iv) Validación y Aseguramiento. A diferencia de un flujo lineal rígido, el modelo incorpora compuertas de decisión estratégica que permiten evaluar la viabilidad técnica antes de la ejecución y verificar el cumplimiento de los criterios adaptativos post-implementación. Asimismo, se establecen ciclos de retroalimentación (*feedback loops*) que garantizan la mejora continua de los componentes arquitectónicos en función de los resultados de validación.

Este instrumento metodológico define no solo *qué* debe transformarse, sino *cómo* ejecutar dicha evolución de manera controlada. Con el fin de proporcionar una guía ejecutable para los equipos de ingeniería, el proceso se desglosa a continuación en sus fases constitutivas, especificando para cada una sus objetivos, los artefactos de entrada (*inputs*), las actividades críticas y los criterios de aceptación (*outputs*) necesarios para validar el avance en el ciclo de vida de la transformación.

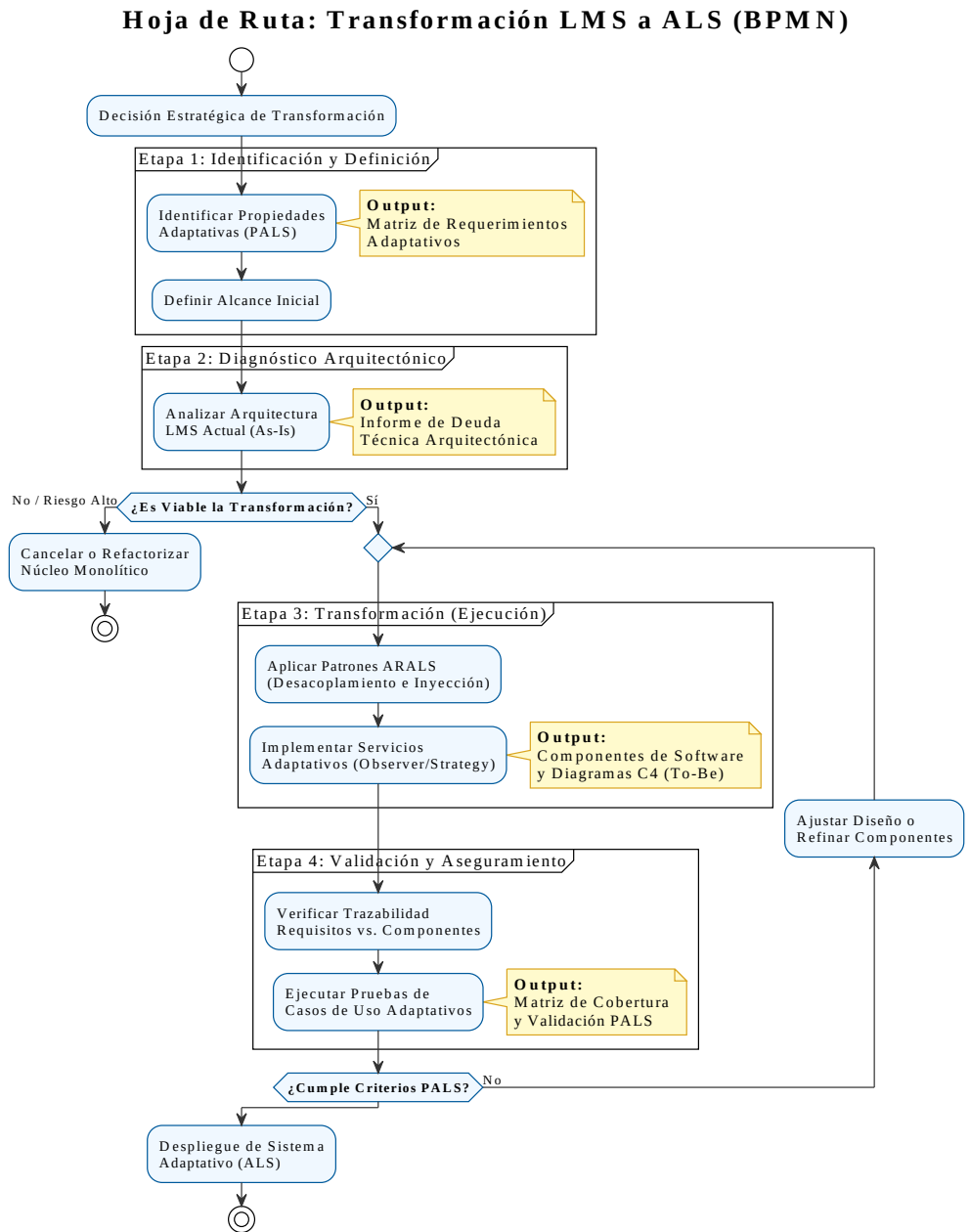


Figura 5.1: Flujo de trabajo de la Hoja de Ruta de transición LMS –> ALS

5.4. Etapas 1 — Identificación de Propiedades Adaptativas y Definición del Punto de Partida de la Transformación

El propósito de la etapa 1 es establecer el punto de partida formal del proceso de transformación arquitectónica del sistema de gestión de aprendizaje (LMS), mediante la elicitación y traducción de requerimientos educativos en propiedades adaptativas (PALS) explícitas. Esta caracterización permite delimitar la brecha funcional y estructural del sistema existente, facilitando la identificación de los componentes arquitectónicos de la referencia ARALS necesarios para soportar la implementación técnica de dichas propiedades.

Esta etapa articula los conceptos teóricos desarrollados en los capítulos precedentes con la realidad del sistema a transformar, constituyendo el nexo metodológico entre la definición abstracta de la adaptatividad y su concreción en artefactos de software verificables. Debido a su relevancia para este trabajo, se define el nexo de la siguiente manera:

Definition 5.2 (Nexo PALS-A $\rightarrow$ PALS). El **nexo PALS-A $\rightarrow$ PALS** se define como el proceso metodológico de traducción técnica mediante el cual las *capacidades estructurales* (formalizadas en el Capítulo 3) se transforman en *instrumentación operativa* (requerimientos funcionales y casos de uso del Capítulo 5). Este nexo actúa como el puente de trazabilidad que permite que la adaptatividad evolucione de una propiedad arquitectónica potencial hacia una funcionalidad verificable y ejecutable a través de la arquitectura ARALS.

La operacionalización de esta fase se fundamenta en la ejecución de una secuencia de actividades diseñadas para garantizar la trazabilidad entre las necesidades pedagógicas y los requerimientos técnicos. Este proceso permite un tránsito sistemático desde la elicitación de objetivos educativos hasta la delimitación precisa de las capacidades estructurales necesarias para la evolución del sistema, estructurándose de la siguiente manera:

1. Derivación de requerimientos adaptativos desde el dominio educativo.

A partir de los objetivos pedagógicos de la institución y de los escenarios de uso previstos, se elicitán los requerimientos funcionales clave vinculados a personalización, trazabilidad, recomendación de contenidos, evaluación del progreso e interoperabilidad. Estos requerimientos representan las capacidades mínimas ne-

cesarias para que el LMS objetivo evolucione hacia un ALS, independientemente de su tecnología base.

2. **Caracterización de Propiedades PALS y Mapeo de Componentes.** Los requerimientos identificados se mapean sistemáticamente con las propiedades del marco PALS (*Properties for Adaptive Learning Systems*). A partir de esta caracterización, se identifican qué componentes arquitectónicos del modelo de referencia ARALS (por ejemplo, motores de inferencia, gestores de perfiles o servicios de monitoreo) son necesarios para satisfacer dichas propiedades. Este paso asegura que la selección de componentes no sea arbitraria, sino que responda a una necesidad adaptativa justificada.
3. **Análisis comparativo *as-is* / *to-be*.** Se construyen escenarios comparativos que contrastan el comportamiento del LMS en su estado original (*as-is*) con el comportamiento esperado bajo la arquitectura de referencia (*to-be*). Este análisis permite visualizar de manera concreta el impacto que tendrán los componentes arquitectónicos seleccionados sobre la experiencia de aprendizaje y la gestión docente.
4. **Identificación de brechas arquitectónicas iniciales.** A partir del contraste entre ambos escenarios, se diagnostican las carencias estructurales de la arquitectura original para soportar los componentes identificados. Se detectan limitaciones tales como la rigidez en el modelo de datos del usuario, la falta de desacoplamiento en la lógica de negocio o la ausencia de mecanismos de captura granular de eventos necesarios para la adaptación.
5. **Definición del alcance inicial de la transformación.** Se delimita el conjunto priorizado de propiedades adaptativas y componentes arquitectónicos que guiarán la intervención sobre el LMS. Este alcance actúa como un contrato técnico que orienta las decisiones de diseño en las etapas posteriores, asegurando que la transformación sea incremental y gestionable.

La culminación de este proceso de identificación y contraste técnico deriva en la consolidación de un conjunto de artefactos y definiciones críticas que constituyen la base de conocimiento para las fases subsiguientes de la hoja de ruta. Los productos tangibles resultantes de esta etapa se detallan a continuación:

- Conjunto explícito de propiedades adaptativas PALS requeridas para el sistema objetivo.

Etapla 2 — Diagnóstico de la Arquitectura *as-is*

- Identificación preliminar de los componentes arquitectónicos de ARALS necesarios para la implementación.
- Escenarios comparativos que evidencian la brecha funcional y estructural entre el LMS actual y el ALS proyectado.
- Definición formal del alcance (*baseline*) de la transformación arquitectónica.

Esta etapa define el *estado inicial controlado* del proceso de transformación y proporciona los insumos fundamentales para la siguiente etapa. Al caracterizar las propiedades y pre-identificar los componentes necesarios, se habilita un diagnóstico técnico preciso en la Etapa 2, garantizando que la evaluación de la arquitectura existente se realice con un foco claro en la viabilidad de soportar los mecanismos adaptativos requeridos.

5.5. Etapla 2 — Diagnóstico de la Arquitectura *as-is*

El propósito de la etapa 2 se centra en la caracterización y análisis de la arquitectura existente del sistema de aprendizaje. Su propósito no es meramente descriptivo, sino diagnóstico: se busca cuantificar el grado de alineación —o la distancia estructural— entre el LMS actual y las propiedades adaptativas definidas en la Etapa 1.

El objetivo central es identificar la "Deuda Técnica Arquitectónica" que impide la implementación inmediata de la adaptatividad. Para ello, se formaliza la siguiente ficha técnica de ejecución:

Concepto	Definición Operativa
Objetivo	Evaluar la viabilidad técnica de la transformación mediante el análisis de brechas estructurales.
Artefactos de Entrada (Inputs)	1. Matriz de Requerimientos Adaptativos (Output de Etapa 1). 2. Código fuente y documentación técnica del LMS legado.
Actividades Clave	1. Recuperación Arquitectónica (<i>Architecture Recovery</i>). 2. Evaluación de Soporte PALS. 3. Análisis de Brechas (<i>Gap Analysis</i>).
Artefactos de Salida (Outputs)	Informe de Deuda Técnica Arquitectónica y Dictamen de Viabilidad.

Tabla 5.1: Ficha técnica de la Etapa 2: Diagnóstico Arquitectónico.

En este caso se definen como actividades operativas cuestiones de diagnóstico que se

ejecutan mediante tres actividades analíticas que profundizan en la estructura interna del sistema:

1. Caracterización mediante Recuperación Arquitectónica

Dado que la documentación de los sistemas legados suele estar desactualizada, se aplica un proceso de ingeniería inversa para abstraer la arquitectura real del LMS. Se identifican los patrones predominantes, típicamente estilos *Layered* o *Model-View-Controller* (MVC) rígidos, caracterizados por:

- **Alto Acoplamiento:** Fuerte dependencia entre la lógica de presentación (vistas) y la lógica de control, dificultando la personalización dinámica.
- **Modelos de Dominio Estáticos:** Entidades centradas en la gestión administrativa (Cursos, Inscripciones) con escasa representación del estado cognitivo o contextual del aprendiz.
- **Mecanismos de Extensión Limitados:** Hooks o plugins orientados a funcionalidad aditiva, no a la modificación del comportamiento del núcleo.

2. Evaluación del Soporte de Propiedades Adaptativas

Se contrasta la arquitectura recuperada con los componentes de ARALS identificados en la Etapa 1. El análisis busca evidencias de las siguientes limitaciones estructurales:

- **Ausencia de Modelo del Aprendiz Dinámico:** Inexistencia de estructuras de datos para almacenar trazas evolutivas (cognitivas, conductuales).
- **Falta de Observabilidad Granular:** Captura de eventos restringida a resultados finales (notas), sin soporte para estándares de trazabilidad continua (como xAPI).
- **Lógica de Negocio Acoplada:** Inexistencia de separación entre la lógica funcional (e.g., mostrar un contenido) y la lógica adaptativa (e.g., decidir *qué* contenido mostrar), lo que impide la inyección de motores de inferencia.

3. Identificación de Brechas y Deuda Técnica

Se consolida el hallazgo de las brechas arquitectónicas que deben ser resueltas. Esta actividad no solo lista los componentes faltantes (e.g., "Falta un Motor de Recomendación"), sino que identifica los riesgos estructurales (e.g., "El modelo de datos no soporta relaciones many-to-many dinámicas").

Etapas 3 — Ejecución de la Transformación Arquitectónica

La etapa concluye con la emisión del **Informe de Deuda Técnica Arquitectónica**. Este documento alimenta la compuerta de decisión crítica del proceso (ver Figura 5.1) según estos indicadores:

- **Viable:** La arquitectura base permite la inyección de dependencias y la extensión de modelos sin refactorizar el núcleo completo. Se procede a la Etapa 3.
- **No Viable (Riesgo Alto):** La rigidez del sistema requiere una reescritura total, desaconsejando la aplicación de la hoja de ruta evolutiva.

Esta etapa constituye el mecanismo de **diagnóstico y viabilidad** del proceso. Su función crítica es objetivar la magnitud de la deuda técnica arquitectónica antes de iniciar la ejecución de cambios. Los hallazgos obtenidos (brechas y limitaciones) alimentan la **compuerta de decisión** del diagrama de flujo, permitiendo dictaminar si la estrategia de transformación evolutiva es factible o si los riesgos estructurales exigen un enfoque alternativo. En caso de proceder, el diagnóstico arquitectónico generado actúa como el insumo directo (*input*) para diseñar la estrategia de intervención en la Etapa 3.

5.6. Etapa 3 — Ejecución de la Transformación Arquitectónica

La etapa 3 constituye la fase de ejecución del procedimiento, donde se materializan las intervenciones de ingeniería planificadas. Su propósito es operacionalizar la transición desde el estado *as-is* (LMS Estándares) hacia el estado *to-be* (Sistemas de Aprendizaje Adaptativo), aplicando los patrones de diseño definidos en la arquitectura de referencia ARALS.

El enfoque no es destructivo, sino evolutivo: se busca inyectar propiedades adaptativas preservando la estabilidad del núcleo transaccional. Para guiar esta ejecución, se define la siguiente ficha técnica:

Hoja de Ruta para la Promoción de Propiedades Adaptativas en Sistemas de Aprendizajes

Concepto	Definición Operativa
Objetivo	Implementar los componentes de ARALS y refactorizar el sistema legado para soportar el ciclo adaptativo.
Artefactos de Entrada (Inputs)	1. Informe de Deuda Técnica y Dictamen de Viabilidad (Etapas 1 y 2). 2. Especificación de Componentes ARALS (Motores, Modelos).
Actividades Clave	1. Desacoplamiento (Patrón Facade/Strangler). 2. Inyección de Servicios Adaptativos (Observer/Strategy). 3. Estandarización de Interoperabilidad.
Artefactos de Salida (Outputs)	Incremento de Software Adaptativo , Modelos de Datos Extendidos y APIs de Integración.

Tabla 5.2: Ficha técnica de la Etapa 3: Transformación Arquitectónica.

La ejecución técnica se rige por principios invariantes que aseguran la calidad del software resultante:

- **Separación de Responsabilidades (SoC):** La lógica de adaptación (reglas pedagógicas) debe estar físicamente desacoplada de la lógica de dominio (gestión de cursos).
- **Incrementalidad (Strangler Fig Pattern):** Las nuevas capacidades adaptativas se construyen alrededor del sistema legado, reemplazando funcionalidades progresivamente para minimizar el riesgo de regresión.
- **Inversión de Dependencias (DIP):** Los módulos de alto nivel (políticas adaptativas) no deben depender de detalles de bajo nivel del LMS, sino de abstracciones.

La transformación se articula en tres líneas de acción arquitectónicas complementarias.

1. **Desacoplamiento Funcional mediante Capas de Servicios** Se introduce una capa intermedia (*Service Layer*) que actúa como barrera de contención ante la complejidad del sistema legado. Utilizando el patrón *Facade*, se abstrae la interacción con el núcleo del LMS, permitiendo:
 - Centralizar la orquestación de flujos de aprendizaje complejos.
 - Interceptar las peticiones del usuario antes de que lleguen al núcleo, habilitando puntos de extensión (*hooks*).

Etapas 3 — Ejecución de la Transformación Arquitectónica

- Reducir el acoplamiento directo entre la interfaz de usuario (Vistas) y la lógica de negocio profunda.
2. Incorporación de Componentes Adaptativos Especializados Se instancian los componentes nucleares de ARALS que materializan el ciclo de control autónomo. Esta actividad implica:
- Implementación de Sensores (Observer Pattern): Se instrumenta el código para capturar eventos granulares (clics, tiempos, errores) sin modificar la lógica transaccional.
 - Motores de Inferencia (Strategy Pattern): Se inyectan algoritmos de recomendación intercambiables que procesan el contexto del estudiante en tiempo real.
 - Modelado Dinámico: Se extienden los esquemas de base de datos para persistir el estado cognitivo y el historial de interacciones (*User Profile Modeling*).
3. Estandarización del Acceso y API Gateway Se define un punto de entrada controlado para los servicios adaptativos, alineado con el patrón API Gateway. Este componente es estratégico para:
- Exponer las capacidades de adaptación (e.g., 'obtener siguiente recomendación') como servicios consumibles por cualquier cliente (web, móvil).
 - Garantizar la interoperabilidad con ecosistemas externos mediante estándares (LTI, xAPI).
 - Facilitar la futura integración de servicios de Inteligencia Artificial externos sin alterar la arquitectura interna.

Rol dentro de la hoja de ruta

Esta etapa representa el *núcleo productivo* del ciclo de vida. Es aquí donde la deuda técnica identificada en la Etapa 2 es gestionada y transformada en capacidad funcional. Los artefactos de software generados en esta fase (código, bases de datos, servicios) constituyen la entrada obligatoria para la Etapa 4, donde se someterán a validación para verificar si, efectivamente, el sistema resultante satisface las propiedades PALS definidas al inicio del proceso.

5.7. Etapa 4 — Validación y Aseguramiento de la Calidad Adaptativa

Esta etapa cierra el ciclo de ingeniería mediante un proceso sistemático de aseguramiento de calidad. Su propósito trasciende el testing funcional tradicional; su objetivo es verificar la *integridad arquitectónica* del sistema transformado. Se busca confirmar empíricamente que la implementación realizada en la Etapa 3 no solo funciona, sino que satisface las propiedades PALS definidas en la Etapa 1 y respeta las restricciones estructurales de ARALS.

Esta fase actúa como el mecanismo de control que habilita o rechaza el despliegue del sistema. Para ello, se establece la siguiente ficha técnica:

Concepto	Definición Operativa
Objetivo	Certificar la alineación entre la implementación técnica, el modelo de referencia ARALS y las propiedades PALS.
Artefactos de Entrada (Inputs)	■ Incremento de Software Adaptativo (Output de Etapa 3). ■ Matriz de Requerimientos Adaptativos (MRA - Etapa 1).
Actividades Clave	1. Auditoría de Trazabilidad (Traceability Check). 2. Verificación Estructural (Architecture Compliance). 3. Evaluación Funcional Comparativa (Delta Testing).
Artefactos de Salida (Outputs)	Matriz de Cobertura PALS y Dictamen de Validación Final.

Tabla 5.3: Ficha técnica de la Etapa 4: Validación y Aseguramiento.

La validación se estructura en tres dimensiones complementarias que triangulan la calidad del sistema.

- Auditoría de Trazabilidad Conceptual Se ejecuta una verificación bidireccional para asegurar que no existan cabos sueltos.^{en} la transformación.
 - Cobertura de Requerimientos: Se verifica que cada propiedad PALS priorizada en la Etapa 1 esté materializada por al menos un componente de software en la Etapa 3.
 - Justificación de Componentes: Se confirma que cada nuevo servicio o modelo introducido responda a una necesidad adaptativa explícita, evitando la sobre-ingeniería (*Gold Plating*).
- Verificación Estructural y Funciones de Aptitud Se evalúa si el sistema respeta los principios arquitectónicos de ARALS, utilizando el concepto de *Fitness*

Etapa 4 — Validación y Aseguramiento de la Calidad Adaptativa

Functions arquitectónicas.

- Verificación de Desacoplamiento: Se comprueba que el núcleo transaccional no tenga dependencias directas hacia los motores de inferencia (violación de capas).
- Integridad de Interfaces: Se valida que los puntos de acceso (APIs) cumplan con los estándares de interoperabilidad definidos, asegurando que el sistema sea abierto y extensible.
- Evaluación Funcional mediante Casos de Uso Comparativos Se realizan pruebas dinámicas contrastando el comportamiento del sistema original (*Baseline*) versus el sistema transformado, enfocándose en el "Delta Adaptativo":
 - Escenarios de Personalización: ¿El sistema altera la ruta de aprendizaje ante un cambio en el perfil del usuario?
 - Respuesta en Tiempo Real: ¿La retroalimentación se genera dinámicamente o sigue patrones estáticos?
 - Observabilidad: ¿Se registran las trazas granulares necesarias para alimentar el ciclo de mejora continua?

La etapa concluye con la generación de la *Matriz de Cobertura PALS*, que alimenta la compuerta de decisión final del diagrama BPMN:

1. *Validado*: El sistema cumple con los criterios de aceptación y las propiedades PALS. Se autoriza el despliegue del ALS.
2. *No Conforme (Feedback Loop)*: Se detectan desviaciones críticas (e.g., latencia excesiva en recomendaciones o fallas en la captura de datos). Se activa el ciclo de retroalimentación hacia la etapa 3 para ajustar los componentes o refinar la implementación.

Esta etapa actúa como el *mecanismo de gobernanza* del proceso de transformación. Su función es garantizar que el producto final no sea simplemente un LMS con funcionalidades añadidas, sino un verdadero Sistema de Aprendizaje Adaptativo arquitectónicamente robusto. Al formalizar los criterios de aceptación, esta etapa cierra la brecha entre la intención pedagógica teórica y la realidad operativa del software.

Capítulo 6

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

6.1. Introducción

En este capítulo se muestra un ejemplo práctico de transformación de un *Learning Management System* (LMS), con el propósito de incorporarle funcionalidades adaptativas, definidas en el Capítulo 3 en la definición 3.1, a través de la transformación controlada de la arquitectura ARALS desarrollada en el capítulo 4, utilizando la guía procedimental del Capítulo 5.

Como caso de estudio se seleccionó el módulo `website_slides` de la aplicación eLearning integrada en la plataforma Odoo.¹ Esta elección se sustenta en los resultados del análisis desarrollado en el Capítulo 3 y su correspondiente hoja de ruta de aplicación de Capítulo 5, donde se examinó la arquitectura interna del módulo y se evidenció su alineamiento con varios de los patrones estructurales y principios de diseño identificados en la caracterización general de los LMS contemporáneos. En particular, el ecosistema modular de Odoo —basado en componentes desacoplados, servicios es-

¹Sitio oficial de Odoo: <https://www.odoo.com>

pecializados y un modelo de extensión incremental mediante addons— ofrece escenario apropiado para evaluar la aplicabilidad de la arquitectura ARALS y de los mecanismos adaptativos propuestos.

Asimismo, el módulo de Odoo `website_slides` constituye un punto de observación privilegiado para estudiar la transición desde un LMS de orientación transaccional hacia un sistema con capacidades adaptativas, dado que presenta elementos clave analizados en el capítulo anterior: gestión estructurada de contenidos, seguimiento de interacciones, infraestructura de eventos, y un modelo de usuario susceptible de enriquecimiento semántico. Estas características permiten mapear con claridad los puntos de variabilidad, las dependencias funcionales y los flujos susceptibles de intervención bajo los criterios establecidos en las Propiedades Adaptativas PALS 3.1.

Finalmente, el hecho de que Odoo sea una plataforma ERP–LMS ampliamente adoptada en entornos corporativos y profesionales refuerza la relevancia del análisis, al permitir evaluar la arquitectura ARALS en un contexto operativo real, donde las decisiones adaptativas deben coexistir con procesos organizacionales transversales.

Odoo es una plataforma de código abierto para aplicaciones empresariales que implementa una arquitectura modular y orientada a modelos. Su diseño promueve la reutilización mediante herencia, composición y extensión no intrusiva de funcionalidades, lo cual lo convierte en un entorno propicio para ensayar mecanismos de transformación arquitectónica sin comprometer el núcleo del sistema. En particular, el módulo `website_slides` representa una implementación funcional de un sistema de gestión de contenidos de aprendizaje (LMS), basado en el patrón Model-View-Controller (MVC) y Microkernel, pero carente de capacidades adaptativas en términos de personalización, análisis de comportamiento o modificación dinámica de la experiencia del usuario.

La elección de este módulo no solo se fundamenta en su estructura abierta y su integración natural con los mecanismos de extensión de Odoo, sino también en su potencial como ejemplo representativo para validar los aportes de esta tesis. El procedimiento propuesto —detallado en el Capítulo 5— permite guiar la evolución del sistema original hacia una arquitectura que incorpore propiedades clave de los *Adaptive Learning Systems*, tales como adaptación de contenidos, modelado del aprendizaje, retroalimentación contextual y analítica de desempeño.

Como punto de partida, el módulo original presenta una estructura orientada principalmente a la entrega de contenidos y actividades de evaluación en una secuencia fija y uniforme para todos los usuarios. Carece de mecanismos para personalizar el recorrido formativo según el perfil del estudiante, ni ofrece herramientas analíticas avanzadas que permitan a docentes monitorear el progreso o intervenir pedagógicamente en forma

proactiva.

La transformación propuesta, en cambio, introduce una arquitectura enriquecida con nuevas capas y servicios especializados. A través de la incorporación de un motor de recomendación, un sistema de modelado del usuario, controladores adaptativos, y una interfaz que se ajusta dinámicamente al contexto del estudiante, se habilita la implementación de rutas de aprendizaje personalizadas y estrategias pedagógicas contextualmente sensibles. Esta evolución se fundamenta en los patrones arquitectónicos definidos en el Capítulo 4, incluyendo: *Layered Architecture*, *Service-Oriented Architecture*, *Context-Aware Services*, y *Rule-Based Decision Modules*.

6.2. Caracterización de los Requerimientos de Mejoras para Odoo Elearning

Los requerimientos funcionales aquí presentados constituyen una derivación sistemática del marco conceptual desarrollado en los capítulos previos, operacionalizando los aportes teóricos sobre *Adaptive Learning Systems* (Capítulo 2), las propiedades PALS 3.1 y la arquitectura ARALS (Capítulo 4). Estos requerimientos representan la concreción funcional de las propiedades adaptativas, actuando como el nexo necesario entre las definiciones abstractas y el plano operativo de la arquitectura objetivo que se implementa en este capítulo.

RQ1 — Generación automática de rutas formativas personalizadas. Este requerimiento establece la capacidad del sistema para gestionar secuencias dinámicas de contenidos, actividades y evaluaciones adaptadas al perfil y desempeño del estudiante. La literatura identifica esta personalización de trayectorias como una evolución de la hipermedia adaptativa, permitiendo abandonar estructuras curriculares rígidas en favor de secuenciaciones flexibles y contextuales sustentadas en reglas pedagógicas y mecanismos de inferencia [5, 21, 133].

RQ2 — Adaptación de la interfaz según el perfil del usuario. Se refiere a la capacidad de modificar dinámicamente la interfaz de usuario en función del rol y contexto del estudiante, tratando a la capa de presentación como un componente activo del proceso adaptativo. Esta capacidad se vincula con los conceptos de *context-awareness*, permitiendo modular la complejidad visual y la organización de la información según las características del usuario, lo cual exige una arquitectura desacoplada que soporte la adaptación en tiempo de ejecución [134, 95, 17].

RQ3 — Trazabilidad y análisis de eventos y recursos de aprendizaje.

Este eje establece la necesidad de capturar y analizar sistemáticamente los eventos de interacción, reconociendo a la trazabilidad como una condición habilitante para la adaptación. La utilización de estándares como xAPI permite estructurar la recolección de datos de manera interoperable, facilitando la auditoría de decisiones adaptativas y proporcionando la base empírica necesaria para los ciclos de retroalimentación del sistema [53, 24, 52].

RQ4 — Recomendación inteligente de contenidos y actividades. Define la capacidad de sugerir recursos que complementen la ruta formativa basándose en el modelo del aprendiz y su desempeño previo. Estos mecanismos de recomendación educativa, integrados con modelos predictivos y técnicas de inteligencia artificial, operan de forma proactiva frente a patrones de dificultad o estancamiento, apoyándose en motores desacoplados para garantizar la escalabilidad y evolución sistémica [14, 100, 103].

RQ5 — Visualización de métricas de desempeño para docentes. Provee visualizaciones avanzadas y contextualizadas que permiten a los docentes interpretar el proceso de aprendizaje individual y colectivo. Este requerimiento es fundamental para la *accountability* del sistema, integrando servicios de analítica y repositorios históricos que transforman los datos trazables en información significativa para la toma de decisiones pedagógicas informadas y la validación de las adaptaciones realizadas [111, 17, 22].

La definición de estos requerimientos marca el pasaje desde la conceptualización de las propiedades PALS hacia su ejecución arquitectónica. Estos cinco ejes guían el proceso de transformación del LMS, actuando como insumo principal para el análisis de brechas (*as-is*), la planificación de transformaciones y la validación final de la solución adaptativa propuesta en las etapas posteriores de la hoja de ruta metodológica.

6.3. Aplicación Etapa 1 — Identificación de Propiedades Adaptativas

La ejecución de la primera etapa del procedimiento metodológico toma como insumo el conjunto de requerimientos adaptativos formalizados en la Sección 6.2. Estos lineamientos definen las necesidades críticas para la evolución del módulo `website_slides`, y abarcan dimensiones estructurales como la adaptación dinámica de la interfaz, la trazabilidad granular y la personalización de las rutas formativas.

Paso 1: Mapeo sistemático entre requerimientos y propiedades PALS

La fundamentación arquitectónica de las necesidades funcionales, necesaria para evitar incorporaciones *ad-hoc*, se materializa a través de una correspondencia formal entre los requerimientos referenciados y las propiedades adaptativas (PALS) del marco teórico. La Tabla 6.1 sistematiza esta relación y delimita el alcance arquitectónico inicial para la transformación del sistema.

Tabla 6.1: Correspondencia entre requerimientos y propiedades PALS

Requerimiento (RQ)	Propiedades PALS asociadas
Adaptación de la interfaz	Personalización dinámica; Context-awareness; Ajuste en tiempo real
Trazabilidad de eventos	Registro granular; Tracking interoperable; Monitoreo continuo
Rutas personalizadas	Adaptación pedagógica; Secuenciación dinámica; Inferencia basada en desempeño
Recomendación de contenidos	Modelado del aprendizaje; Análisis predictivo; Selección inteligente de recursos
Métricas docentes	Analítica de aprendizaje; Detección de riesgo; Retroalimentación contextualizada
Interoperabilidad externa	Integración estandarizada; Conectividad distribuida; Persistencia unificada

A partir de este mapeo, se identifican como componentes críticos de ARALS para esta intervención: el *AdaptiveLearningService* (Capa de Aplicación), el *RecommendationEngine* (Capa de Adaptación) y el *xAPI Tracker* (Capa de Datos).

Paso 2: Casos de uso comparativos

Siguiendo la lógica procedimental establecida, se procede a contrastar el comportamiento funcional del sistema mediante la especificación de casos de uso. Este ejercicio tiene por propósito evidenciar la brecha operativa entre el módulo original (*as-is*) y la versión transformada, demostrando cómo las propiedades adaptativas alteran el flujo de interacción y la lógica de negocio.

En primer lugar, la Tabla 6.2 describe la interacción estándar en el módulo original. En este escenario, la navegación es estática y el sistema actúa como un mero repositorio de contenidos, sin intervención pedagógica automática.

En contraste, la Tabla 6.3 presenta los nuevos casos de uso habilitados por la arquitectura ARALS. Aquí se observa la intervención de los servicios de personalización

Aplicación Etapa 2 – Análisis de la Arquitectura *as-is*

Elemento	CU-00 Consumir contenido estándar
Objetivo	Permitir que el estudiante acceda a los contenidos del curso en el orden predeterminado del canal sin personalización.
Actor	Estudiante inscrito en el curso.
Precondición	El estudiante está autenticado. Cuenta con acceso al canal y a las diapositivas públicas del mismo.
Flujo	1. El estudiante ingresa al curso desde el portal. 2. El sistema muestra todas las diapositivas del canal según la secuencia estándar. 3. El estudiante selecciona una diapositiva cualquiera y la consume. 4. El sistema registra el avance básico (por ejemplo, marcado como completado en el historial general de slides). 5. El estudiante repite el proceso con otras diapositivas sin recibir sugerencias específicas.
Postcondición	El estudiante ha consumido diapositivas del canal sin que se genere ni actualice una ruta personalizada.
Notas	No existe filtrado por nivel de dificultad ni priorización adaptativa. El canal muestra el mismo contenido para todos los estudiantes. Las métricas se limitan al historial general del curso.

Tabla 6.2: Caso de uso previo a la implementación de rutas adaptativas

(generación de rutas) y de seguimiento avanzado (monitoreo), lo cual valida la incorporación efectiva de las propiedades de *Personalización Adaptativa Colaborativa (PAC)* y *Evaluabilidad Adaptativa*.

La comparación entre el escenario previo descrito en la Tabla 6.2 y los nuevos flujos detallados en la Tabla 6.3 cristaliza las diferencias funcionales: mientras el primero depende de la iniciativa exclusiva del usuario sobre una estructura rígida, los segundos introducen al sistema como un agente activo que orquesta la experiencia de aprendizaje en función del contexto.

El contraste de los casos de uso evidencia la carencia de un motor de inferencia y de un registro de eventos granular en la arquitectura original. En consecuencia, se define el **baseline de la transformación** focalizado exclusivamente en la propiedad PAC (Rutas de Aprendizaje), postergando integraciones multi-plataforma externas por considerarse fuera del alcance de los objetivos de esta implementación.

6.4. Aplicación Etapa 2 – Análisis de la Arquitectura *as-is*

La segunda etapa consiste en el análisis sistemático de la arquitectura existente del módulo `website_slides` de Odoo eLearning. Este paso se alinea con el objetivo general

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

Elemento	CU-01 Generar ruta	CU-02 Monitorear ruta
Objetivo	Crear o actualizar automáticamente una ruta de aprendizaje adaptada al estudiante.	Proveer al estudiante una vista consolidada del avance y contenidos.
Actor	Estudiante inscrito.	Estudiante con ruta activa.
Precondición	Estudiante autenticado. Existen diapositivas configuradas.	Ruta activa existente. Historial de progreso disponible.
Flujo	1. Estudiante ingresa al curso. 2. Sistema pide ruta activa. 3. Servicio crea ruta consultando motor de recomendaciones. 4. Ruta se actualiza filtrando lo completado. 5. Interfaz muestra opciones.	1. Estudiante abre tablero. 2. Sistema calcula progreso. 3. Genera instantánea de estado. 4. Interfaz despliega lista. 5. Estudiante marca avance y actualiza historial.
Postcondición	Ruta activa vinculada y ordenada.	Visión consolidada y sincronizada.
Notas	Visibilidad restringida al propietario. Recálculo al cambiar dificultad.	Recálculo ante cambios en historial. Notificaciones por umbrales.

Tabla 6.3: Casos de uso de rutas de aprendizaje para el módulo modificado a través de la arquitectura ARALS

del método: contrastar las capacidades reales del sistema base frente a las propiedades adaptativas identificadas en la Etapa 1, evaluando su grado de soporte, ausencia o incompatibilidad estructural. El análisis *as-is* constituye, por tanto, el insumo crítico para fundamentar las transformaciones arquitectónicas posteriores.

Desde una perspectiva funcional, el módulo `website.slides` implementa un sistema de gestión de contenidos educativos que organiza cursos, lecciones y evaluaciones siguiendo una lógica secuencial y homogénea para todos los usuarios. Su arquitectura responde al modelo modular típico de Odoo, basado en *addons*, modelos ORM, vistas web, controladores HTTP y mecanismos de renderizado orientados a la presentación. Esta estructura resulta adecuada para escenarios de aprendizaje estático, pero no contempla mecanismos necesarios para la variación dinámica ni para la personalización basada en el contexto del estudiante.

La contrastación de la arquitectura original con los requerimientos funcionales derivados en la Etapa 1 —particularmente aquellos vinculados a la personalización de la interfaz, la trazabilidad granular, la recomendación y la visualización del progreso— permite corroborar una de las premisas centrales de esta tesis: la viabilidad de las implementaciones funcionales adaptativas se encuentra subordinada a la existencia de estructuras arquitectónicas específicas que las habiliten. En este sentido, el análisis no solo revela carencias operativas, sino que evidencia limitaciones estructurales de fondo

que impiden el despliegue de las propiedades PALS.

En primer término, se observa una ausencia de un modelo dinámico del aprendiz, dado que el sistema original no registra patrones de comportamiento ni mantiene un perfil actualizado del usuario que permita inferir necesidades pedagógicas o estados cognitivos específicos. Asimismo, existe una trazabilidad limitada donde los eventos se restringen a métricas transaccionales básicas, como el acceso o la completitud de una lección, careciendo de soporte para estándares de monitoreo granular como *xAPI*, indispensables para un ciclo adaptativo basado en evidencia.

Se detecta también una falta de mecanismos de adaptación en tiempo de ejecución, provocando que la navegación y la presentación de contenidos sigan rutas fijas y uniformes para todos los estudiantes, sin capacidad de ajuste dinámico basado en el desempeño en tiempo real o en las preferencias detectadas. La arquitectura presenta un acoplamiento interno elevado, donde la lógica de presentación, la gestión de contenido y la evaluación se encuentran rígidamente integradas en los controladores. Esta estructura dificulta el desacoplamiento necesario para incorporar motores de recomendación o servicios analíticos externos de forma modular y escalable.

Finalmente, se evidencia una limitada integración inteligente, ya que la plataforma no provee interfaces normalizadas para conectar motores externos de inferencia o modelos de inteligencia artificial, quedando cualquier intento de personalización restringido a extensiones manuales y atómicas sobre el código base. Estas observaciones se sintetizan en la Tabla 6.4, donde se comparan los componentes clave de la arquitectura original con los componentes requeridos por la arquitectura adaptativa propuesta. Como se evidencia, la brecha entre ambos modelos abarca aspectos funcionales, estructurales y de interoperabilidad.

Tabla 6.4: Comparativa Arquitectura Modelo y Odoo eLearning

Elemento	Arq. Adaptativa	Arq. Odoo eLearning	Mejoras necesarias
Web App / Mobile UI	Presentación adaptada por perfil	JS.UI, plantillas no adaptativas	Integrar <i>Adaptive UI Engine</i>
Dashboard Estudiante	Vista personalizada según desempeño	No presente	Incorporar dashboard basado en métricas y progreso
Dashboard Docente	Análisis granular por usuario/grupo	No presente	Agregar panel docente para monitoreo pedagógico
Adaptive UI Engine	Componente modular de adaptación	No presente	Implementar motor para re-configuración dinámica
API Gateway	Capa explícita de desacoplamiento	Implícita en controladores	Introducir <i>API Gateway</i> para servicios adaptativos

La contrastación estructural evidencia que, si bien la arquitectura base ofrece una plataforma robusta para la gestión transaccional de contenidos, resulta insuficiente para soportar propiedades adaptativas complejas como la personalización dinámica o la inferencia pedagógica. La rigidez inherente al patrón MVC monolítico —donde la lógica de control está fuertemente acoplada a la presentación y al modelo de datos— impide la orquestación flexible de rutas de aprendizaje y limita la capacidad de respuesta ante el contexto del usuario.

Estas limitaciones estructurales se corresponden directamente con la ausencia de los mecanismos definidos en las propiedades PALS (Capítulo 5), validando la necesidad de una intervención profunda guiada por la arquitectura de referencia ARALS (Capítulo 4). No se trata, por tanto, de una mera extensión funcional, sino de una refactorización necesaria para habilitar capacidades que el diseño actual inhibe.

En síntesis, el análisis *as-is* permite formalizar la brecha técnica entre el módulo existente y el sistema objetivo. Esta distancia arquitectónica se organiza en torno a tres ejes de transformación que guiarán la Etapa 3 del procedimiento:

La instrumentación del sistema (Capa de Monitoreo) implica la transición de un registro de estados finales a una captura granular de eventos, permitiendo alimentar el modelo del aprendiz con datos precisos de interacción.

El desacoplamiento de la lógica adaptativa (Capa de Servicios) se centra en la extracción de las reglas de negocio desde los controladores web hacia servicios especializados, rompiendo así la dependencia directa entre la vista y el modelo.

Finalmente, la incorporación de **inteligencia (Motores de Inferencia)** consiste en la integración de componentes de recomendación y perfilado que operen sobre los datos capturados para modificar el comportamiento del sistema en tiempo de ejecución.

A partir del diagnóstico sintetizado en la Figura 6.1, se concluye con un Dictamen de Viabilidad Positivo. La capacidad de extensión no intrusiva de Odoo permite una transformación evolutiva factible sin reescribir el núcleo, habilitando formalmente el inicio de la fase de ejecución de la hoja de ruta.

6.5. Aplicación Etapa 3 – Transformación Arquitectónica

La tercera etapa del procedimiento metodológico (ver Sección 5.6) se centra en la reconfiguración estructural del módulo. A partir del diagnóstico de brechas realizado en la Etapa 2, se diseña la arquitectura objetivo (*to-be*) instanciando los componentes

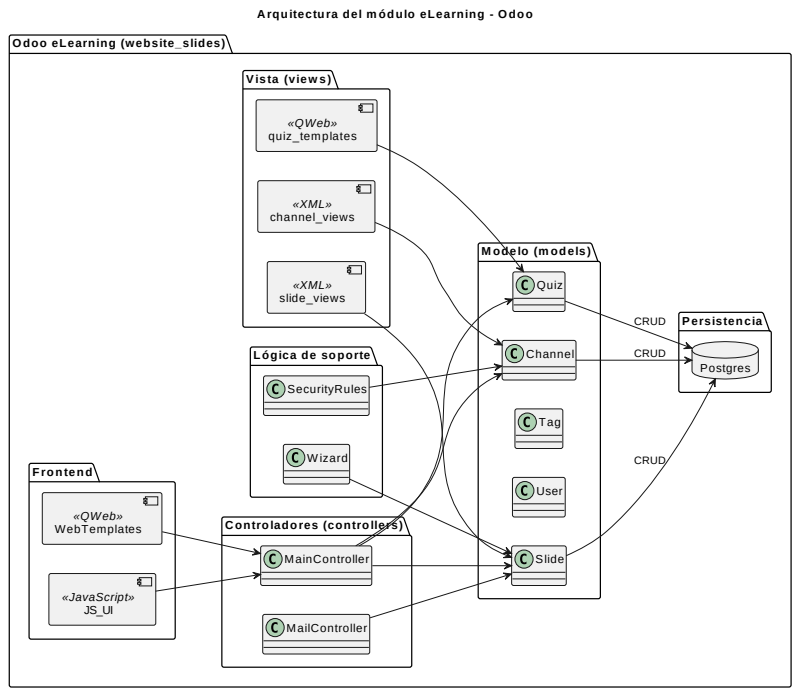


Figura 6.1: Arquitectura *as-is* del módulo eLearning de Odoo: Punto de partida para el Diagnóstico de Deuda Técnica y análisis de brechas estructurales (Etapa 2)

de referencia de ARALS sobre la plataforma Odoo. Esta intervención se ejecutó bajo el patrón *Strangler Fig*, permitiendo envolver la funcionalidad transaccional de Odoo con nuevas capacidades adaptativas sin alterar el código base del núcleo.

El diseño base del módulo `website_slides`, ilustrado en la Figura 6.1, responde a un patrón *Model-View-Controller (MVC)* monolítico. Como se observa en el diagrama, los componentes de la capa de *Controladores* (e.g., `MainController`) mantienen una relación de dependencia directa y fuerte hacia los Modelos de datos (`Slide`, `Channel`, `Quiz`).

Esta disposición presenta una limitación estructural crítica para la adaptatividad: la lógica de navegación y presentación está "incrustada" en el controlador. No existe un mecanismo intermedio que intercepte la petición del usuario para evaluar su perfil o contexto antes de consultar a la base de datos Postgres. El flujo es transaccional y directo (*Vista* → *Controlador* → *Modelo*), lo que impide la inyección de estrategias pedagógicas dinámicas sin modificar intrusivamente el código fuente original.

Para superar estas limitaciones, se aplicaron las tres líneas de acción definidas

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

en el método, dando lugar a la arquitectura extendida presentada en la Figura 6.2. A continuación, se detallan los componentes agregados y la justificación de sus relaciones:

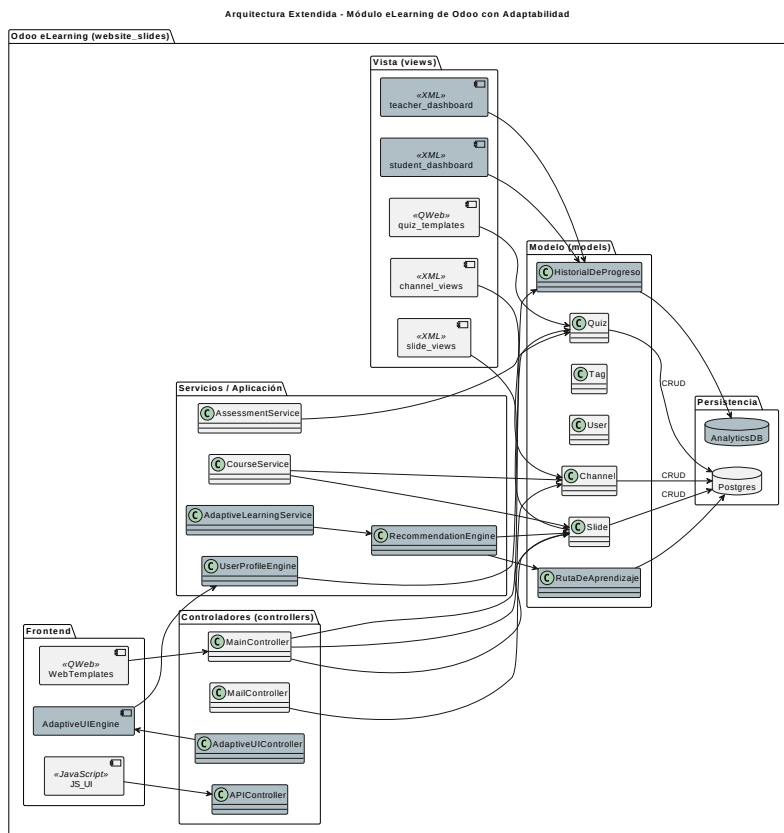


Figura 6.2: Arquitectura *to-be* (Modelo C4): Incremento de Software Adaptativo mediante la inyección de la Capa de Servicios y desacoplamiento de componentes (Etapa 3).

En primera instancia, se procedió al desacoplamiento mediante una capa de Servicios (*Service Layer*), introduciendo una nueva capa lógica para romper la dependencia directa del MVC original. El componente agregado fundamental es el `AdaptiveLearningService`, el cual asume la responsabilidad de orquestar la totalidad de la lógica de negocio adaptativa. Esta intervención genera un cambio en las relaciones funcionales: el controlador ya no invoca directamente al modelo `Slide` para determinar el siguiente contenido, sino que delega dicha decisión al servicio mediante el método `get_recommendations()`, cumpliendo así con el principio de desacoplamiento funcional establecido en la metodología.

Posteriormente, se realizó la Instrumentación mediante Modelos Especializados, extendiendo el modelo de datos para soportar la persistencia de trazas y estados cognitivos que eran inexistentes en el esquema original. Los componentes agregados, específicamente `Slide.Historial.Progreso` y `Slide.Ruta.Aprendizaje`, materializan técnicamente los componentes T1 (Monitorización) y T2 (Modelado) de la arquitectura ARALS. Estos nuevos modelos permiten capturar y almacenar la granularidad de eventos necesaria (bajo el estándar xAPI) que la estructura original de `Postgres` no permitía registrar. En cuanto a su relación operativa, estos modelos son alimentados de forma continua por el `AnalyticsService` aplicando el patrón Observer, lo que garantiza que la recolección de datos no penalice el rendimiento del flujo principal de navegación.

Finalmente, la integración del `RecommendationEngine` y el `AdaptiveUIController` mediante los patrones *Strategy* y *Facade* operacionaliza la inferencia y el acceso unificado al sistema. Esta implementación, al simplificar la comunicación entre el *frontend* y la lógica de personalización, constituye el *Incremento de Software Adaptativo* definido en la hoja de ruta.

6.6. Aplicación Etapa 4 – Validación de la Transformación Arquitectónica

Esta etapa verifica la consistencia y efectividad de la evolución del módulo `website.slides`, demostrando que la transición del estado *as-is* al *to-be* constituye una evolución estructural alineada con ARALS y las propiedades PALS. La validación se articula en tres dimensiones:

Dimensión 1: Trazabilidad Conceptual. Se confirma la correspondencia entre el espacio del problema (RQ1–RQ6) y la solución implementada en Odoo, garantizando soporte arquitectónico para cada requerimiento. Propiedades como la Personalización Adaptativa Colaborativa y la Evaluabilidad Adaptativa se materializan en artefactos tangibles: la monitorización (T1) en el modelo `slide.historial.progreso` y la inferencia (T3) en el `RecommendationEngine`. Los nuevos componentes, como el `AdaptiveLearningService`, responden a necesidades adaptativas explícitas, asegurando la integridad conceptual del diseño.

Dimensión 2: Verificación Estructural respecto a ARALS. Se ratifica la calidad técnica mediante el cumplimiento de patrones de referencia. El desacoplamiento funcional se logra aislando la inferencia en la Capa de Servicios Adaptativos

bajo el patrón *Layered Architecture*. Asimismo, la separación de responsabilidades entre observación (**AnalyticsService**), decisión (**RecommendationEngine**) y ejecución (**AdaptiveUIController**) respeta el ciclo MAPE-K. Finalmente, el uso de fachadas y esquemas compatibles con estándares de trazas en `slide.ruta.aprendizaje` asegura la interoperabilidad en entornos empresariales reales. Esta verificación actuó como una función de aptitud (*fitness function*), confirmando que el núcleo de Odoo permanece agnóstico a la lógica de adaptación.

Dimensión 3: Evaluación Funcional Comparativa. La ejecución de los casos de uso CU-01 y CU-02 evidencia un cambio cualitativo hacia el dinamismo, donde las rutas varían según el desempeño registrado en `slide.historial.progreso`. La reactividad contextual permite remediaciones automáticas vía **AdaptiveLearningService** ante eventos como fallos en cuestionarios. Además, los dashboards analíticos otorgan visibilidad al proceso, transformando el LMS en un sistema transparente que expone la lógica de las recomendaciones y dota al entorno de las capacidades adaptativas objetivo. La correlación de estas pruebas permite consolidar la **Matriz de Cobertura PALS**, verificando que cada propiedad ha sido instrumentada y validada operativamente.

En síntesis, la validación confirma que la transformación es conceptualmente trazable, estructuralmente sólida y funcionalmente efectiva. En este contexto, esta verificación actuó como una función de aptitud (*fitness function*), confirmando que el núcleo de Odoo permanece agnóstico a la lógica de adaptación. En consecuencia, los resultados corroboran que la hoja de ruta propuesta permite evolucionar un LMS legado hacia un sistema adaptativo sin comprometer su estabilidad ni mantenibilidad, fundamentando la generalización del método.

6.7. Detalles de la implementación técnica y estructura de componentes

Esta sección documenta la materialización técnica de la arquitectura extendida sobre el módulo `website_slides` de Odoo. A diferencia de las secciones previas, que abordaron el diseño conceptual, aquí se detallan las decisiones de ingeniería de software, la organización de ficheros y los patrones de diseño aplicados para transformar un esquema MVC tradicional en una arquitectura orientada a servicios adaptativos, cumpliendo con la Etapa 4 de la metodología.

El punto de partida es la estructura estándar del módulo Odoo eLearning (Figura

Detalles de la implementación técnica y estructura de componentes

ra 6.3). Este diseño responde a un patrón *Model-View-Controller* estricto, donde la lógica de negocio reside en los modelos ORM (carpeta `models/`) y la lógica de presentación se distribuye entre controladores HTTP (`controllers/`) y vistas XML (`views/`). Si bien esta organización asegura la coherencia transaccional, carece de capas intermedias para gestionar la complejidad algorítmica requerida por la adaptatividad.

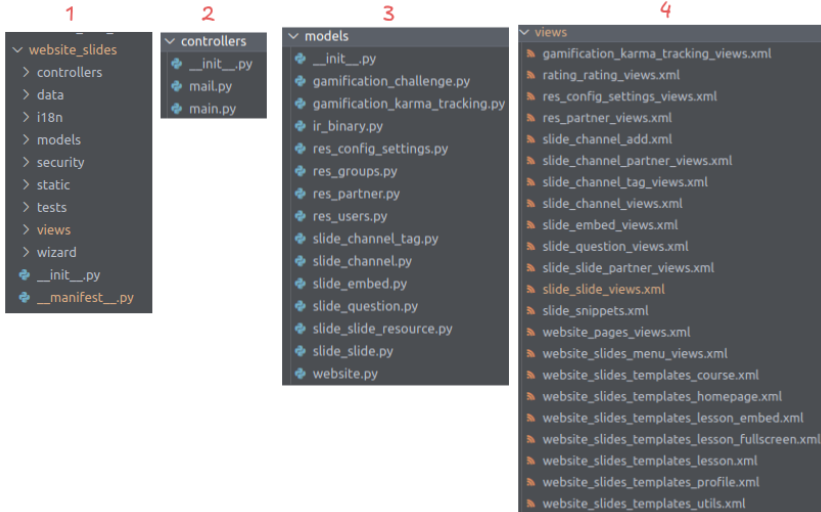


Figura 6.3: Organización de componentes en la arquitectura base (*as-is*): Evidencia de la rigidez estructural previa a la transformación

Para implementar la arquitectura ARALS sin romper la compatibilidad con el núcleo de Odoo (principio *Open/Closed* de SOLID), se optó por una estrategia de extensión no intrusiva. La Figura 6.4 ilustra la nueva estructura de directorios resultante de la transformación. Se destaca la aparición de una Capa de Servicios (implementada conceptualmente mediante modelos abstractos y clases de servicio en Python) y la segregación de la lógica adaptativa en archivos específicos.

Las principales modificaciones estructurales se articularon en tres ejes fundamentales que garantizan la extensibilidad del sistema. En primer término, se desarrollaron los Modelos de Instrumentación mediante la creación de archivos como `models/slide_historial_progreso.py`, diseñados específicamente para la captura de trazas xAPI de manera independiente a los modelos de contenido originales. Complementariamente, se incorporaron los Motores de Inferencia a través de componentes lógicos, tales como `modelsrecommndation_engine.py`, que operan como servicios puros y se encuentran desacoplados de la persistencia directa de datos para favorecer la

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

escalabilidad. Finalmente, la capa de presentación fue intervenida mediante Vistas Extendidas utilizando archivos XML dedicados (`viewswebsite_slide_adaptive.xml`), los cuales permiten la inyección dinámica de componentes en la interfaz de usuario empleando mecanismos de herencia y `xpath`.

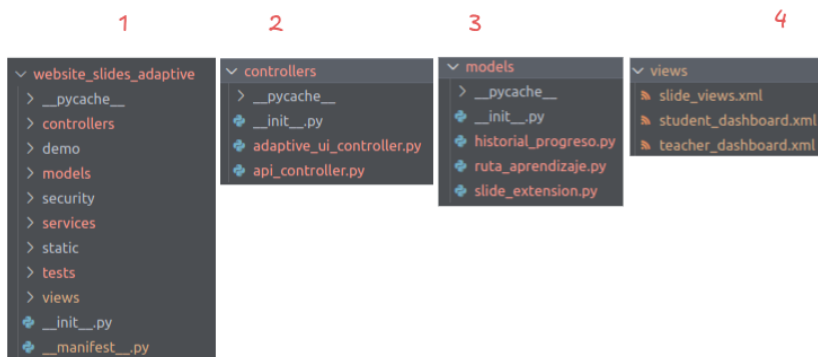


Figura 6.4: Materialización del desacoplamiento funcional y la creación de la capa de servicios adaptativos (Etapa 3).

6.7.1. Implementación de Patrones y Componentes Críticos

La transformación arquitectónica no se limita a añadir archivos, sino que introduce patrones de diseño que alteran el flujo de control del LMS. A continuación, se describen las implementaciones técnicas más relevantes por capa.

Para habilitar la propiedad de *Evaluabilidad Adaptativa*, fue necesario superar la limitación del modelo de datos original, que solo registraba estados finales (aprobado/-desaprobado). Se implementó el modelo `slide.historial.progreso` bajo un esquema de Patrón Observer. Este modelo “observa” las interacciones del usuario y registra granularmente el tiempo y el desempeño, sin afectar el flujo transaccional principal.

```
1
2 # Metadatos para el algoritmo de recomendacion
3 nivel_dificultad = fields.Selection([
4     ('basico', 'Basico'),
5     ('intermedio', 'Intermedio'),
6     ('avanzado', 'Avanzado')
7 ], string='Nivel Cognitivo', default='basico')
```

Listing 6.1: Instrumentación técnica del modelo de datos: Extensión de metadatos para el soporte de las propiedades PALS (Etapa 3).

Detalles de la implementación técnica y estructura de componentes

Strategy La innovación central de la implementación es la introducción de la *Service Layer*. En lugar de que el controlador web decida qué contenido mostrar (lógica *hard-coded*), delega esta decisión al `RecommendationEngine`. Este componente implementa el Patrón Strategy, permitiendo conmutar entre diferentes algoritmos de recomendación (e.g., basado en contenido o filtrado colaborativo) sin modificar el código del controlador.

```
8
9 def get_recommendations(self, user, current_slide=None):
10     """
11     Invoca al motor de recomendación aplicando la estrategia
12     definida en la configuración del curso.
13     """
14     profile = self.env['user.profile.engine'].get_profile_vector(user)
15     strategy = self._get_recommendation_strategy()
16     return strategy.compute(profile, current_slide)
```

Listing 6.2: Desacoplamiento funcional: Implementación de la Capa de Servicios mediante el orquestador de lógica adaptativa (Etapa 3).

En la capa de controladores (`controllers/`), se refactorizó la lógica para actuar como una Fachada (Facade). El controlador `AdaptiveUIController` simplifica la interacción del frontend: recibe la petición HTTP, consulta a la capa de servicios y devuelve una estructura de datos lista para ser renderizada, ocultando la complejidad del cálculo de rutas.

A nivel de interfaz de usuario (Vistas QWeb), se utilizó el Patrón Template Method mediante la directiva `xpath`. Esto permite inyectar fragmentos de interfaz adaptativa (como barras de progreso personalizadas o etiquetas de dificultad) en puntos específicos de las plantillas originales, sin reescribirlas.

```
1 <record id="view_slide_adaptive_detail" model="ir.ui.view">
2     <field name="name">slide.adaptive.detail</field>
3     <field name="inherit_id" ref="website_slides.slide_fullscreen"/>
4     <field name="arch" type="xml">
5         <xpath expr="//div[@class='o_wslides_lesson_info']" position="
6             before">
7             <div t-if="slide.nivel_dificultad" class="badge badge-info">
8                 <field name="nivel_dificultad"/>
9             </div>
10         </xpath>
11     </field>
12 </record>
```

Listing 6.3: Inyección de componentes en el frontend: Extensión no intrusiva de la interfaz de usuario basada en el contexto del aprendiz (Etapa 3).

La implementación detallada evidencia cómo la abstracción arquitectónica de ARALS se traduce en código ejecutable. Mediante el uso de herencia, inyección de dependencias (vía el registro de Odoo) y separación de responsabilidades en servicios, se ha logrado transformar un módulo monolítico en un sistema capaz de soportar las propiedades PALS. Para el equipo de desarrollo, esto implica un cambio de paradigma: pasar de programar transacciones CRUD a diseñar orquestaciones de servicios, donde la lógica de adaptación reside en componentes puros y testables, garantizando la mantenibilidad y escalabilidad del sistema a largo plazo.

6.8. Contribuciones funcionales para la Experiencia de Usuario del caso de uso

Esta sección detalla las contribuciones de ARALS en Odoo eLearning mediante la comparación de una versión estándar frente a una adaptativa, evidenciando cómo la arquitectura propuesta potencia la personalización pedagógica. Mientras que la versión estándar emplea una estructura lineal con materiales y navegación fijos que limitan la retroalimentación a mensajes genéricos, este enfoque estático carece de la capacidad de respuesta necesaria para atender las diferencias individuales del alumnado.

En contraste, la versión ARALS integra los componentes `UserProfileEngine` y `RecommendationEngine` para dinamizar el aprendizaje basándose en el perfil y desempeño del estudiante. Tras un diagnóstico inicial, el sistema adapta en tiempo de ejecución recursos multimedia, la dificultad de los cuestionarios y ofrece diapositivas de refuerzo ante errores, permitiendo evaluar el impacto técnico de la adaptatividad sintetizado en la Tabla 6.5.

En este marco, la Figura 6.5 ilustra las interfaces de usuario incorporadas como parte de la extensión ARALS y permite visualizar de qué manera estos principios arquitectónicos se traducen en nuevos flujos operativos dentro del módulo eLearning. La figura muestra cómo las funcionalidades relacionadas con la configuración, selección y seguimiento de las hojas de ruta se integran directamente en la experiencia del estudiante y del docente, complementando y materializando las mejoras pedagógicas sintetizadas previamente en la Tabla 6.5.

Contribuciones funcionales para la Experiencia de Usuario del caso de uso

Tabla 6.5: Comparación de materiales didácticos entre Odoo eLearning estándar y versión ARALS

Tipo de material	Odoo eLearning estándar	Odoo eLearning con ARALS
Documento PDF	Guía básica sobre el tema	Documento PDF adaptado al perfil del usuario
Video explicativo	Video general para todos los estudiantes	Video seleccionado según estilo de aprendizaje (visual/auditivo)
Infografía	Infografía fija y uniforme	Infografía adaptada al perfil cognitivo
Quiz	Preguntas fijas de opción múltiple	Quiz adaptativo con dificultad variable y feedback enriquecido
Slide teórico	Contenido textual uniforme	Contenido ajustado a preferencias lector/visual/auditivo
Slide multimedia	Imagen o video estándar	Contenido multimedia seleccionado dinámicamente por RecommendationEngine
Slide reforzador	No disponible	Activado cuando el usuario presenta fallos recurrentes
Recomendaciones	No disponible	Sugerencias personalizadas según desempeño del usuario
Diagnóstico inicial	No disponible	Cuestionario inicial para activar el UserProfileEngine
Certificación	Certificado único al finalizar el curso	Certificación + ruta de aprendizaje personalizada sugerida

El formulario de acceso (1) y el menú de configuración (2) constituyen la entrada a las funciones adaptativas de ARALS, habilitando la *ruta de aprendizaje personalizada* frente a la rigidez del LMS estándar. Estas interfaces gestionan diagnósticos y recomendaciones iniciales, permitiendo desplegar contenido diferenciado según el perfil cognitivo del estudiante y vinculando la operatividad del sistema con la Tabla 6.5.

La hoja de ruta (3) y la incorporación de recursos (5) representan el núcleo donde se definen materiales y secuencias. En esta etapa se materializa la transición del catálogo estático a uno dinámico, permitiendo al docente configurar PDFs adaptados, videos por preferencia cognitiva o quizzes de dificultad ajustable. Cada recurso añadido constituye una decisión pedagógica operativa que responde directamente al desempeño detectado en el aprendiz.

Por su parte, la selección de usuario (4) y el filtrado de recursos (6) consolidan el diagnóstico inicial para vincular perfiles específicos a trayectorias personalizadas. Mediante estas interfaces, los docentes filtran contenidos por estilo de aprendizaje o nivel de complejidad, facilitando la activación de instancias críticas como el “Video explicativo adaptado” o el “Slide reforzador” según las necesidades identificadas por el motor de inferencia.

Finalmente, los indicadores de progreso (7) y el porcentaje de avance (8) emplean el `HistorialDeProgreso` para ofrecer un seguimiento detallado en lugar de estimaciones

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

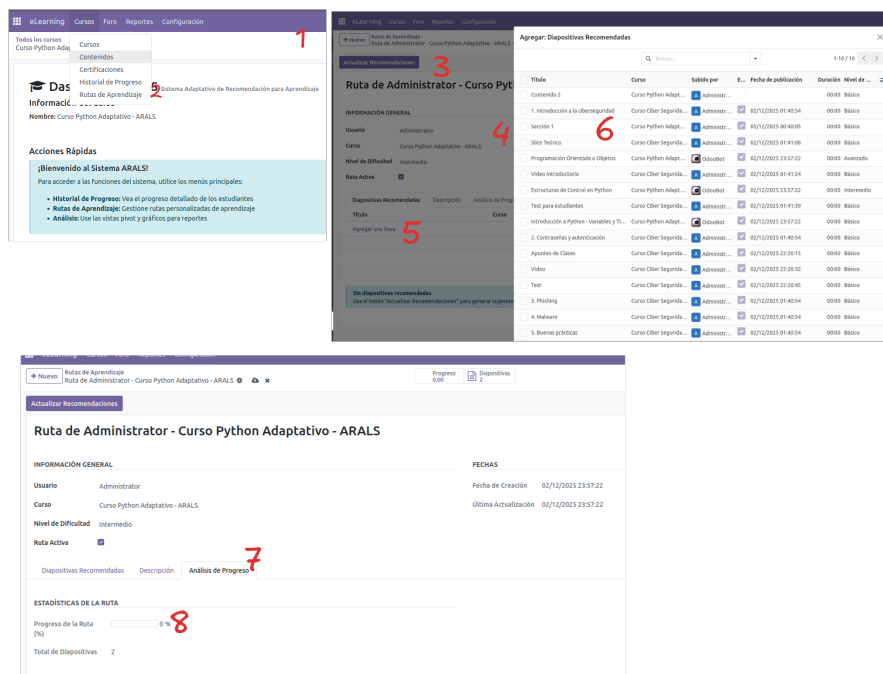


Figura 6.5: Interfaces de configuración adaptativa: Instrumentación funcional de la propiedad PAC y el motor de perfiles de usuario (Etapa 4)

genéricas. El módulo `tesis_website_slide_arals` integra estas herramientas en el flujo de Odoo (Fig. 6.6), proporcionando los paneles adaptativos fundamentales para operacionalizar los mecanismos de personalización introducidos por la arquitectura ARALS.

La Figura 6.7 presenta el *Tablero de Rutas de Aprendizaje* para la gestión de itinerarios personalizados (nivel, recursos y progreso), funcionalidad que se complementa con el *Tablero de Progreso* (Figura 6.8), donde se sintetizan métricas de seguimiento dinámico como el tiempo dedicado y la puntuación obtenida. Finalmente, la Figura 6.9 muestra la perspectiva del estudiante, integrando la ruta adaptativa en la interfaz de contenidos para ofrecer recursos priorizados y orientación personalizada; de este modo, el sistema operacionaliza el ciclo adaptativo de ARALS al vincular la gestión docente con la experiencia final del usuario.

El *Tablero de Rutas de Aprendizaje* (Figura 6.7) permite gestionar itinerarios personalizados mediante niveles y recursos, complementándose con el *Tablero de Progreso* (Figura 6.8) para sintetizar métricas de seguimiento dinámico como el tiempo y la



Figura 6.6: Interfaz de navegación extendida: Punto de acceso a los servicios adaptativos e instrumentación de las propiedades PALS en el frontend (Etapa 3)

puntuación. Asimismo, la Figura 6.9 integra la ruta adaptativa en la interfaz del estudiante, proporcionando recursos priorizados que operacionalizan el ciclo de ARALS al vincular la gestión docente con la experiencia del usuario. En conjunto, estas interfaces constituyen evidencia visual de la implementación realizada y demuestran la traducción efectiva de la arquitectura en un módulo funcional que amplía sustantivamente las capacidades pedagógicas de Odoo.

La validación práctica de ARALS se complementó con un video demostrativo que exhibe la configuración y actualización dinámica de rutas de aprendizaje personalizadas². La secuencia (00:00) contrasta la experiencia de un administrador y un estudiante sobre un mismo curso, evidenciando la estructura diferenciada que habilita ARALS frente al modelo estático original. El núcleo del proceso (00:23) reside en la capacidad de asignar componentes específicos a usuarios individuales, permitiendo al administrador (00:36) gestionar recursos mediante los nuevos formularios especializados de la arquitectura.

A través de un ejemplo concreto (00:49), se observa la asignación de materiales adaptados (v.g. refuerzos o infografías según perfil) a un estudiante específico. La posterior verificación inmediata (01:25) confirma que el nuevo contenido es visible únicamente en la vista del usuario destino, ratificando la independencia total de itinerarios entre perfiles (01:42). La demostración concluye (02:01) subrayando que esta personalización dinámica es un desarrollo original de esta investigación doctoral, inexistente en la versión nativa de Odoo eLearning. En conjunto, el flujo audiovisual evidencia la viabilidad técnica y pedagógica de ARALS para transformar un LMS convencional en un sistema adaptativo centrado en el usuario.

²Enlace al video: <https://kommodo.ai/recordings/rn0F09WSGLEXPu6BHcl?tab=transcript>

Aplicación del Modelo Arquitectónico Propuesto en un Caso de Estudio: Odoo eLearning

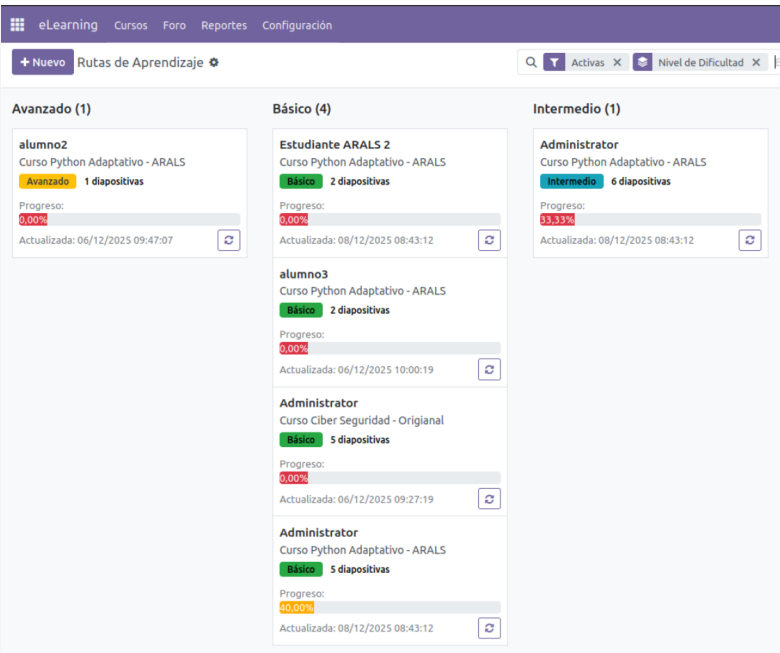


Figura 6.7: Validación funcional de rutas adaptativas: Ejecución de la secuenciación dinámica basada en el desempeño del aprendiz (Etapa 4)

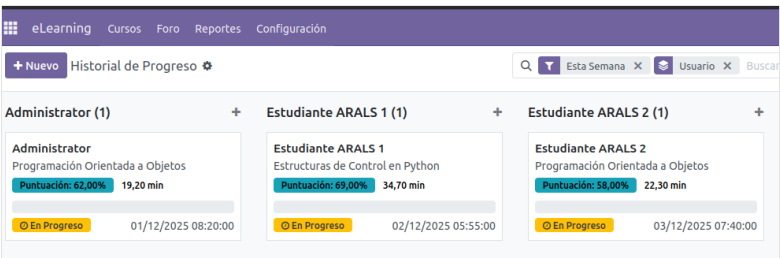


Figura 6.8: Visualización de métricas de progreso: Verificación de la Propiedad de Retroalimentación Adaptativa y monitoreo continuo (Etapa 4)

Contribuciones funcionales para la Experiencia de Usuario del caso de uso



Figura 6.9: Interfaz adaptativa del estudiante: Validación de la Personalización Adaptativa Colaborativa (PAC) y ejecución de rutas dinámicas (Etapa 4)

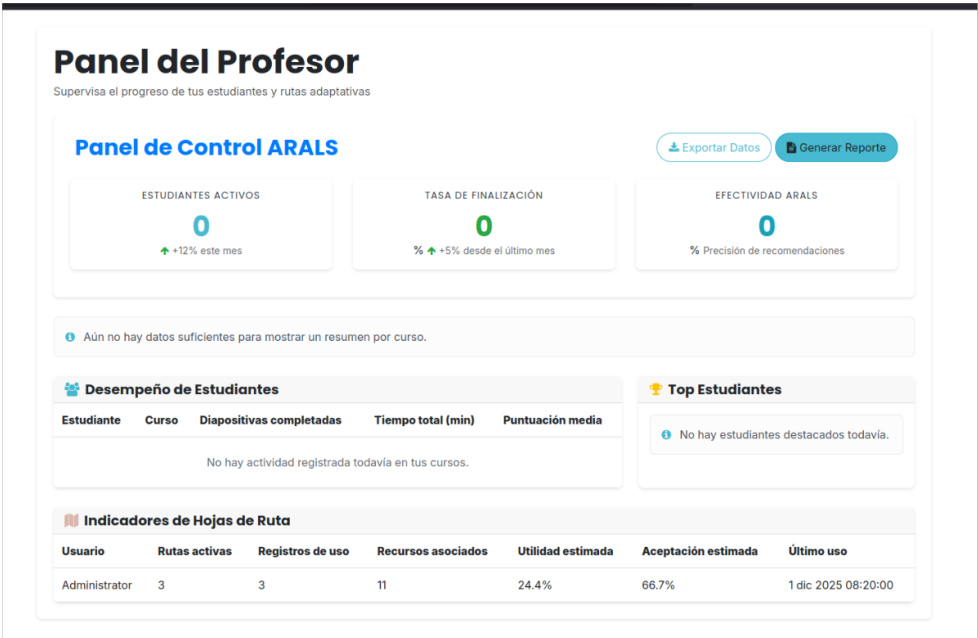


Figura 6.10: Tablero analítico para el docente: Operacionalización de la Propiedad Multistakeholder y transparencia algorítmica (Etapa 4)

Capítulo 7

Discusión, Contribuciones y Conclusiones

El presente capítulo constituye el cierre de la investigación doctoral, integrando los hallazgos obtenidos a lo largo de las fases teóricas, metodológicas y prácticas del estudio. Tras haber establecido los fundamentos de los Sistemas de Aprendizaje Adaptativo (Capítulo 3), diseñado la arquitectura de referencia ARALS (Capítulo 4), definido una hoja de ruta de transformación (Capítulo 5) y validado su viabilidad técnica en el entorno Odoo (Capítulo 6), corresponde ahora realizar una síntesis crítica de estos esfuerzos.

El propósito de este capítulo es triple: primero, discutir los resultados obtenidos contrastándolos con la hipótesis inicial y el estado del arte; segundo, sistematizar las contribuciones originales de la tesis detallando su impacto en la ingeniería de software educativo; y tercero, exponer con honestidad intelectual las limitaciones del estudio y las líneas de investigación futura que de él se derivan.

7.1. Discusión de Resultados

La discusión de los resultados se articula en torno a la validación de la hipótesis principal de esta tesis, la cual sostenía que es posible dotar de capacidades adaptativas a un LMS tradicional mediante una intervención arquitectónica no destructiva, basada en un modelo de referencia.

El análisis comparativo entre el módulo `website_slides` original (monolítico) y la

versión transformada con ARALS (modular-adaptativa) confirma que la rigidez de los LMS no es una propiedad intrínseca del software, sino una deuda técnica arquitectónica. Los resultados del Capítulo 6 evidencian que, al desacoplar la lógica de presentación de la lógica de negocio mediante la introducción de una *Capa de Servicios Adaptativos*, se logra romper el determinismo de la navegación lineal. Esto valida la premisa teórica de que la adaptatividad es, fundamentalmente, un problema de diseño estructural antes que de sofisticación algorítmica.

Contrario a enfoques que sugieren el reemplazo total de plataformas heredadas por nuevos sistemas nativos (*Greenfield*), esta investigación demuestra la viabilidad del enfoque evolutivo (*Brownfield*). La implementación en Odoó probó que mecanismos como la herencia de modelos, la inyección de dependencias y el patrón *Observer* permiten "instrumentar" un sistema existente para capturar trazas granulares (Propiedad PALS de Monitorización) sin comprometer su estabilidad. Esto discute y complementa la literatura existente sobre modernización de software, aportando un caso de éxito específico para el dominio educativo.

A diferencia de los modelos puramente teóricos analizados en el Capítulo 2, ARALS ha demostrado su capacidad de instanciación práctica. Mientras que muchas arquitecturas de referencia en e-learning permanecen en un nivel de abstracción alto, la validación funcional mediante los casos de uso (rutas dinámicas y recomendación contextual) confirma que la arquitectura propuesta logra aterrizar conceptos abstractos (como el ciclo MAPE-K) en componentes de software tangibles y operativos.

7.2. Sistematización de Contribuciones

Más allá de la validación de la hipótesis, esta investigación genera un conjunto de aportes originales que enriquecen el cuerpo de conocimiento de la Ingeniería de Software aplicada a la Educación. A continuación, se detallan estas contribuciones, clasificándolas según su naturaleza y alcance.

Uno de los primeros hallazgos de esta tesis fue la dispersión conceptual existente en torno al término 'adaptatividad' en el software educativo. Como respuesta, se ha sintetizado y formalizado el marco de propiedades **PALS** (*Properties for Adaptive Learning Systems*). Este aporte trasciende la mera definición pedagógica para establecer requerimientos no funcionales de software verificables. Al definir propiedades como la *Evaluabilidad Adaptativa*, la *Trazabilidad Granular* y la *Inferencia Contextual*, la tesis provee un vocabulario técnico unificado que permite a ingenieros evaluar capacidades reales. Su utilidad quedó demostrada al servir como base para el diseño de los

modelos de datos extendidos (e.g., *HistorialDeProgreso*) implementados en el caso de estudio.

El núcleo central de la investigación reside en el diseño y especificación de **ARALS** (*Architecture Reference for Adaptive Learning Systems*), consistente en una arquitectura de referencia canónica que integra el ciclo de control autonómico (MAPE-K) dentro de una estructura de capas (*Layered Architecture*). Esta arquitectura resuelve el problema del acoplamiento entre la lógica de dominio y la lógica de adaptación, ofreciendo una plantilla reutilizable para la evolución de sistemas legados. La validación en Odoo demostró la eficacia de este diseño al permitir la inserción de un *RecommendationEngine* desacoplado y controladores adaptativos sin alterar el núcleo transaccional del ERP.

Reconociendo que la mayoría de las instituciones educativas operan sobre plataformas heredadas, esta tesis no se limitó a proponer una arquitectura ideal, sino un camino para alcanzarla mediante la definición de un **Procedimiento Metodológico de Transformación** de cuatro etapas (Análisis, Diagnóstico, Transformación y Validación). Este método constituye una herramienta de ingeniería para la gestión de la deuda técnica y provee una guía prescriptiva para refactorizar sistemas monolíticos. Su aplicación permitió identificar con precisión qué componentes nativos de Odoo debían extenderse y qué nuevos servicios debían crearse, evitando desarrollos ad-hoc y garantizando la sostenibilidad.

La viabilidad de los conceptos anteriores se demostró mediante su instanciación técnica, logrando la implementación exitosa de los componentes de ARALS sobre el módulo `website_slides` de Odoo. Esta prueba de concepto (PoC) evidencia que es técnicamente posible inyectar inteligencia adaptativa en un sistema transaccional estándar. El artefacto de software resultante, disponible en repositorio abierto, sirve como referencia práctica para la comunidad académica y profesional al demostrar la operacionalización de patrones de diseño como *Observer*, *Strategy* y *Facade*.

7.3. Limitaciones del Trabajo

La validez de los resultados presentados debe interpretarse considerando las siguientes limitaciones inherentes al alcance de la investigación:

Dependencia del Caso de Estudio: Si bien ARALS se plantea como una arquitectura de referencia agnóstica, la validación empírica se ha realizado exclusivamente sobre la plataforma Odoo. Aunque Odoo es representativo de sistemas MVC monolíticos (como Moodle o Sakai), la generalización de los resultados a arquitecturas

radicalmente diferentes (e.g., microservicios nativos o serverless) requiere validaciones adicionales para confirmar la universalidad del modelo.

Sofisticación Algorítmica: El foco de la tesis ha sido eminentemente *arquitectónico* (el contenedor) y no *algorítmico* (el contenido). Los motores de recomendación e inferencia implementados en el caso de estudio utilizan reglas heurísticas y estadísticas descriptivas en lugar de modelos avanzados de Deep Learning o Redes Neuronales. No obstante, la arquitectura de referencia queda diseñada y preparada para soportar la integración de dichos modelos en futuras iteraciones.

Evaluación Longitudinal: La validación funcional se realizó mediante pruebas de sistema y casos de uso controlados que permitieron verificar la operatividad técnica. Sin embargo, no se llevó a cabo un despliegue en producción con una cohorte masiva de estudiantes a lo largo de un ciclo académico completo, lo que limita la capacidad de este estudio para medir el impacto pedagógico a largo plazo o el rendimiento del sistema bajo condiciones de alta concurrencia real.

7.4. Líneas Futuras de Investigación

A partir de los hallazgos y limitaciones expuestos, se identifican las siguientes líneas para la continuidad de la investigación:

Validación en Múltiples Plataformas: Se plantea la necesidad de replicar el procedimiento metodológico propuesto en otros LMS populares de código abierto, como Moodle o Chamilo. Este esfuerzo permitirá refinar el modelo ARALS y verificar su universalidad, asegurando que los principios de transformación arquitectónica sean aplicables y efectivos en diversos entornos tecnológicos educativos.

Incorporación de IA Avanzada: Resulta estratégico aprovechar la *Capa de Servicios Adaptativos* ya creada para integrar servicios externos de Inteligencia Artificial, incluyendo modelos de lenguaje de gran escala (LLMs) y motores de recomendación basados en filtrado colaborativo. El objetivo de esta línea es mejorar sustancialmente la precisión de las sugerencias y la capacidad de inferencia del sistema frente al comportamiento del usuario.

Estandarización de Interoperabilidad: Es fundamental profundizar en la implementación de estándares internacionales como LTI (Learning Tools Interoperability) y especificaciones xAPI completos dentro de la capa de integración de ARALS. Esta acción permitirá la federación de datos entre distintas instituciones y plataformas, garantizando una comunicación fluida y estandarizada dentro del ecosistema de aprendizaje adaptativo.

7.5. Reflexión Final

La culminación de esta investigación doctoral invita a una reflexión que trasciende los aspectos puramente técnicos de la arquitectura de software o las métricas de rendimiento del caso de estudio. Al inicio de este trabajo, la premisa fundamental sostenía que la rigidez de los Sistemas de Gestión del Aprendizaje (LMS) constituía una barrera para la personalización educativa. Tras el desarrollo del marco ARALS y su validación empírica, esta premisa no solo se confirma, sino que revela que la denominada rigidez arquitectónica” (architectural stiffness) de las plataformas tradicionales es el verdadero obstáculo para superar el paradigma de ”talla única”. La arquitectura de software, por tanto, no es un contenedor neutro, sino la condición de posibilidad de la pedagogía digital.

Durante años, la brecha entre las aspiraciones de las teorías del aprendizaje y la realidad de las plataformas tecnológicas se ha intentado cerrar mediante la acumulación de contenidos o la sofisticación de interfaces. Sin embargo, esta tesis demuestra que la limitación era estructural y sistémica. La introducción del marco PALS (Properties for Adaptive Learning Systems) y la Secuencia de Transformación Conceptual ha permitido tender un puente formal entre las necesidades pedagógicas y los requerimientos de ingeniería. Sin una arquitectura subyacente diseñada bajo el ciclo autónomo MAPE-K para monitorear, modelar y actuar, cualquier intento de personalización es cosmético. En este sentido, el aporte más significativo de este recorrido es evidenciar que la adaptatividad es, ante todo, un problema de diseño estructural que requiere el desacoplamiento efectivo de la lógica de adaptación del núcleo transaccional del sistema.

Asimismo, este trabajo pone en valor el concepto de sostenibilidad tecnológica y gestión de la deuda técnica en el ámbito educativo. En un contexto donde la obsolescencia digital es la norma, la metodología propuesta para transformar sistemas legados (brownfield) adquiere una dimensión estratégica. Demostrar que una plataforma monolítica como Odoos puede evolucionar hacia un ecosistema inteligente mediante una estructura de capas desacopladas, sin ser destruida ni reemplazada, ofrece un camino viable para que instituciones con infraestructuras heredadas puedan modernizarse de forma incremental. La innovación no siempre requiere construir desde cero; el desafío más valioso reside en dotar de nueva inteligencia a las herramientas operativas que ya sostienen el proceso educativo.

La trayectoria de esta investigación evidencia que el futuro del software educativo depende de la capacidad de poner la lógica computacional al servicio de la necesidad

Reflexión Final

humana de aprender. Si bien la tecnología avanza hacia horizontes dominados por la Inteligencia Artificial generativa, la lección final de este estudio permanece vigente: ninguna inteligencia artificial puede operar eficazmente sobre una estructura rígida y sorda. La arquitectura es el sistema nervioso que permite a la tecnología educativa cobrar vida, convirtiendo un repositorio estático en un compañero adaptativo del aprendizaje humano

Bibliografía

- [1] Neil Selwyn. On the limits of artificial intelligence (AI) in education. *Nordisk tidsskrift for pedagogikk og kritikk*, 10(1):3–14, 2024. Ensayo crítico sobre las limitaciones de la IA en contextos pedagógicos.
- [2] Ben Williamson. *The Future of the Curriculum: School Knowledge in the Digital Age*. MIT Press, Cambridge, MA, 2013. Analiza el impacto de los medios digitales en el conocimiento oficial y el currículo escolar.
- [3] Marc Alier, María José Casany, and Jordi Piguillem. Moodle 2.0: Shifting from a learning toolkit to an open learning platform. In *Knowledge Management, Information Systems, E-Learning, and Sustainability Research (WSKS 2010)*, volume 111 of *Communications in Computer and Information Science*, pages 1–10, Berlin, Heidelberg, 2010. Springer.
- [4] Moodle Community. Moodle architecture documentation. <https://moodledev.io/general/docs/architecture>, 2023. Consultado: 23 de enero de 2026. Documentación técnica oficial sobre la infraestructura de Moodle.
- [5] Peter Brusilovsky and Eva Millán. User models for adaptive hypermedia and adaptive educational systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 3–53. Springer-Verlag, Berlin, Heidelberg, 2007.
- [6] Alexandros Paramythis and Susanne Loidl-Reisinger. Adaptive learning environments and e-learning standards. *Electronic Journal of e-Learning*, 2(1):181–194, 2004.
- [7] Peter Brusilovsky. Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2):87–110, 2001.
- [8] Sabine Graf, Kinshuk, Qingsheng Zhang, Paul Maguire, and Victoria Shtern. An architecture for dynamic student modelling of learning styles in learning systems and its application for adaptivity. pages 103–110, 2010.
- [9] Elisa Yumi Nakagawa et al. Consolidating a process for the design, representation, and evaluation of reference architectures. *2011 WICSA/ECSA Joint Working Conference on Software Architecture*, 2011.

Bibliografía

- [10] Samuil Angelov, Paul Grefen, and Danny Greefhorst. A classification of software reference architectures: Analyzing their success and effectiveness. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*, pages 141–150, Cambridge, UK, 2009. IEEE.
- [11] Lev S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.
- [12] Barry J. Zimmerman. Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41(2):64–70, 2002.
- [13] John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285, 1988.
- [14] Hendrik Drachsler, Hans G. K. Hummel, and Rob Koper. Personal recommender systems for learners in lifelong learning networks: The requirements, techniques and model. *International Journal of Learning Technology*, 3(4):404–423, 2008.
- [15] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 4th edition, 2021.
- [16] Nick Rozanski and Eóin Woods. *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley Professional, Upper Saddle River, NJ, 2nd edition, 2011.
- [17] Aldair Ruiz Nepomuceno, Eduardo López Domínguez, Saúl Domínguez Isidro, Martín Alberto Magaña Nieto, Amilcar Meneses-Viveros, and Jorge de la Calleja. Software architectures for adaptive mobile learning systems: A systematic literature review. *Applied Sciences*, 14(11):4540, 2024.
- [18] Abbas Fadhil Aljuboori and Hala Al-lawati. Intelligent adaptive e-learning systems: Current approaches, architectures, and applications. *European Conference on e-Learning*, 22(1):1925–1934, 2023.
- [19] Inés Dussel. *Algoritmos, Escuela y Justicia Educativa*. Siglo XXI Editores, Buenos Aires, Argentina, 2023.
- [20] Vija Vagale, Laila Niedrite, and Svetlana Ignatjeva. The architecture of the personalized adaptive e-learning system. *Applied Computer Systems*, 23(2):87–95, 2018.
- [21] Vija Vagale, Laila Niedrite, and Svetlana Ignatjeva. Implementation of personalized adaptive e-learning system. *Baltic Journal of Modern Computing*, 8(2):315–334, 2020.
- [22] Ekerete Ibam, Olumide Sunday Adewale, Osemwegie Christopher Agbonifo, and Ifeoma M. Akindeji. Modeling of an adaptive e-learning system for improved learning performance. *International Journal of Information System and Computer Science*, 7(1):45–62, 2023.

- [23] IMS Global Learning Consortium. Ims learning tools interoperability specification. Technical report, IMS Global Learning Consortium, 2010. Especificación LTI para integración de herramientas.
- [24] Advanced Distributed Learning (ADL). Experience api (xapi) specification. Technical report, Advanced Distributed Learning Initiative, 2013. Especificación xAPI para tracking de experiencias.
- [25] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, Boston, MA, 3rd edition, 2012.
- [26] R. Kazman, M. Klein, and P. Clements. Atom: Method for architecture evaluation (cmu/sei-2000-tr-004). Technical report, SEI/CMU, 2000.
- [27] M. Barbacci, R. Ellison, T. Lattanze, J. A. Stafford, C. Weinstock, and W. Wood. Quality attribute workshops (qaws). Technical report, SEI/CMU, 2001.
- [28] ISO/IEC. Systems and software engineering – systems and software quality requirements and evaluation (square) – system and software quality models. International Standard ISO/IEC 25010:2011, International Organization for Standardization, Geneva, Switzerland, 2011.
- [29] ISO/IEC. Systems and software engineering – systems and software quality requirements and evaluation (square) – measurement of system and software product quality. International Standard ISO/IEC 25023:2016, International Organization for Standardization, Geneva, Switzerland, 2016.
- [30] ISO/IEC/IEEE. Software, systems and enterprise – architecture description. International Standard ISO/IEC/IEEE 42010:2022, International Organization for Standardization / Institute of Electrical and Electronics Engineers, Geneva, Switzerland, 2022.
- [31] ISO/IEC/IEEE. Software, systems and enterprise – architecture processes. International Standard ISO/IEC/IEEE 42020:2019, International Organization for Standardization / Institute of Electrical and Electronics Engineers, Geneva, Switzerland, 2019.
- [32] Odysseas Tsilingiridis and Alexandros Karakasidis. Milms: A microservices-based learning management system. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4314–4321. IEEE, 2020.
- [33] Khalil Dahri, Muhammad Ali Akbar, and Kashif Mehmood. Migration from monolithic to microservices architecture in lms: A systematic approach. *IEEE Access*, 9:78326–78341, 2021.
- [34] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.

Bibliografia

- [35] David Garlan, Shang-Wen Cheng, An-Cheng Huang, Bradley Schmerl, and Peter Steenkiste. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, 37(10):46–54, 2004.
- [36] Yong Jiang. Construction of learning management system based on sakai. *Journal of Xuchang University*, 2008.
- [37] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, and Judith Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley Professional, 2nd edition, 2010.
- [38] IMS Global Learning Consortium. Ims learning design specification. Technical report, IMS Global Learning Consortium, 2003.
- [39] Neal Ford et al. *Building Evolutionary Architectures*. 2017.
- [40] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. O’Reilly Media, Sebastopol, CA, 1st edition, 2015. Existe 2ª edición (2021): ISBN 978-1-492-03402-5.
- [41] ISO/IEC/IEEE. Software, systems and enterprise – architecture description. International Standard ISO/IEC/IEEE 42010:2022, International Organization for Standardization / Institute of Electrical and Electronics Engineers, Geneva, Switzerland, 2022.
- [42] IEEE Standards Association. Ieee 7000-2021 - ieee standard model process for addressing ethical concerns during system design. Technical report, IEEE, 2021.
- [43] R. Kazman, M. Klein, and P. Clements. *ATAM: Method for Architecture Evaluation*. Carnegie Mellon Software Engineering Institute, Pittsburgh, PA, 2000.
- [44] Norbert Rudolf Busch and Andrzej Zalewski. A systematic literature review of enterprise architecture evaluation methods. *ACM Computing Surveys*, 57(5):1–36, 2025.
- [45] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. PublicAffairs, New York, NY, 2019. Análisis crítico del capitalismo de vigilancia.
- [46] Wayne Holmes and Kaśka Porayska-Pomsta. *The Ethics of Artificial Intelligence in Education: Practices, Challenges, and Debates*. Routledge, London, UK, 2022.
- [47] Rafael Capilla et al. Capturing software architecture knowledge for pattern-driven design. In *Proceedings of the 5th European Conference on Software Architecture*. Springer, 2009.
- [48] Danny Weyns, Sam Malek, and Jesper Andersson. Forms: Unifying reference model for formal specification of distributed self-adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems*, 7(1):8:1–8:61, 2012.

- [49] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. 2021.
- [50] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin, Germany, 3rd edition, 2019.
- [51] The Open Group. *TOGAF Standard, Version 9.2*. The Open Group, Reading, UK, 2018.
- [52] Ryan S. Baker and Aaron Hawn. Algorithmic bias in education. *International Journal of Artificial Intelligence in Education*, 32(4):1052–1092, 2022.
- [53] George Siemens and Ryan SJD Baker. Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10):1380–1400, 2013.
- [54] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.
- [55] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- [56] Alejandro Sartorio and Maximiliano Cristiá. Primera aproximación al diseño e implementación de los DHD. In *XXXIV Conferencia Latinoamericana de Informática (CLEI)*, Santa Fe, Argentina, 2008.
- [57] Alejandro Sartorio. First approximation to DHD design and implementation. *CLEI Electronic Journal*, 17(1):1–11, 2014.
- [58] Alejandro R. Sartorio, Soledad Ayala, and Alejandro Hernandez. Diseño de aplicaciones e-learning adaptativas: superando limitaciones mediante arquitecturas modulares y escalables. In *Jornadas Argentinas de Informática e Investigación Operativa (JAIIO)*, Argentina, 2025. Universidad Nacional de La Plata.
- [59] Alejandro R. Sartorio, Soledad Ayala, and Alejandro L. Hernandez. Open gateway de servicios de inteligencia artificial generativa para la gestión de expedientes digitales en la administración pública. In *Jornadas Argentinas de Informática e Investigación Operativa (JAIIO)*, Argentina, 2024. Universidad Nacional de La Plata.
- [60] Alejandro R. Sartorio. Una arquitectura conceptual para implementar algoritmos de inteligencia artificial en la administración pública. In *Simposio de Informática en el Estado (SIE)*, Argentina, 2023. Universidad Nacional de La Plata.
- [61] Soledad Ayala, Alejandro R. Sartorio, and Alejandro Hernandez. Educación superior, plataformas e inteligencia artificial (ia): un acercamiento a los estudios previos. In *Jornadas Argentinas de Informática e Investigación Operativa (JAIIO)*, Argentina, 2025. Universidad Nacional de La Plata.

Bibliografía

- [62] Alejandro Sartorio and Gustavo Rossi. From learning management systems to adaptive learning systems: A systematic architectural transformation framework. In *Proceedings of the XXIX Ibero-American Conference on Software Engineering (CIBSE 2026)*, Caxias do Sul, Brasil, 2026. Aceptado para publicación / Peer-reviewed.
- [63] David Lorge Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15, 1972.
- [64] Robert C Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017.
- [65] Robert C Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008.
- [66] Bertrand Meyer. *Object-Oriented Software Construction*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1997. Classic text on object-oriented design principles.
- [67] Martin L Abbott and Michael T Fisher. *The Art of Scalability*. Addison-Wesley Professional, 2009.
- [68] Bennet P Lientz, E Burton Swanson, and Gloria E Tompkins. Software maintenance management. *The Journal of Systems and Software*, 1:232–240, 1980.
- [69] Dickson Chen, Guy Doumeingts, and Francois Vernadat. Interoperability in collaborative computing environments. *Computers in Industry*, 59(9):859–868, 2008.
- [70] Mikael Svahnberg, Jilles van Gurp, and Jan Bosch. A taxonomy of variability realization techniques. *Software: Practice and Experience*, 35(8):705–754, 2005.
- [71] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 2nd edition, 2018.
- [72] Glenn E Krasner and Stephen T Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming*, 1(3):26–49, 1988.
- [73] Chris Richardson. *Microservices Patterns: With Examples in Java*. Manning Publications, Shelter Island, NY, 2018.
- [74] Rick Kazman, Mark Klein, and Paul Clements. Atam: Method for architecture evaluation. In *Proceedings of the 19th International Conference on Software Engineering (ICSE)*, pages 54–63. ACM, 1998.
- [75] Benjamin S Bloom. The 2 sigma problem. *Educational Researcher*, 13(6):4–16, 1984.

- [76] Barry J Zimmerman. Self-regulated learning and academic achievement. *Educational Psychologist*, 25(1):3–17, 2000.
- [77] John R Anderson. *An Integrated Theory of the Mind*. Oxford University Press, 2004.
- [78] Jean-Paul Doignon and Jean-Claude Falmagne. *Knowledge Spaces*. Springer, 1999.
- [79] Albert T Corbett and John R Anderson. Knowledge tracing. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.
- [80] John H Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1992.
- [81] Francis Heylighen. *The Science of Self-Organization and Adaptivity*. EOLSS Publishers, 2001.
- [82] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, 2009.
- [83] Steven Johnson. *Emergence*. Scribner, 2001.
- [84] Eduardo F Camacho and Carlos Alba Bordons. *Model Predictive Control*. Springer, 2nd edition, 2013.
- [85] Karl Johan Åström and Björn Wittenmark. *Adaptive Control*. Dover Publications, 2nd edition, 2013.
- [86] Kemin Zhou, John C Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, 1996.
- [87] Betty HC Cheng, Rogério de Lemos, et al. Software architecture for self-adaptive systems: Research challenges. *Software Engineering for Self-Adaptive Systems*, pages 1–26, 2009.
- [88] Jeff Kramer and Jeff Magee. *Self-managed systems: an architectural challenge*. 2007.
- [89] Brendan Burns, Joe Beda, and Kelsey Hightower. *Monitoring Distributed Systems*. O’Reilly Media, 2018.
- [90] Kerry Rodden, Hilary Hutchinson, and Xin Fu. Measuring the user experience on a large scale. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010.
- [91] Paul De Bra, Geert-Jan Houben, and Hongjing Wu. Aham: a dexter-based reference model for adaptive hypermedia. *Proceedings of the tenth ACM Conference on Hypertext and hypermedia*, 1999.
- [92] Hyacinth S Nwana. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4):251–277, 1990.

Bibliografia

- [93] IEEE Learning Technology Standards Committee. Ieee standard for learning technology—learning technology systems architecture (ltsa). Technical report, IEEE Computer Society, 2001.
- [94] Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, et al. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372(n71), 2021.
- [95] George Wamamu Musumba and Ruth Wario. Towards a context-aware adaptive e-learning architecture. In *International Conference on Learning and Collaboration Technologies*, pages 234–248, 2018.
- [96] Konstantinos Tsilingeridis and Theodoros Karakasidis. Modular microservices architecture for scalable learning management systems. *International Journal of Educational Technology in Higher Education*, 17(1), 2020.
- [97] Khalil Dahri, Muhammad Ali Akbar, and Kashif Mehmood. Migration from monolithic to microservices architecture in lms: A systematic approach. *IEEE Access*, 9:78326–78341, 2021.
- [98] Parvin Rejina and Sanjay Kumar Dwivedi. Cloud-native architecture for adaptive learning management systems. In *2020 International Conference on Computer Communication and Informatics*, pages 1–6. IEEE, 2020.
- [99] Wei Chen, Xiaoming Liu, and Yifei Zhang. Deep neural network model for student performance prediction using learning analytics. *Computers & Education*, 142:103–115, 2019.
- [100] Meet Joshi. Adaptive learning through artificial intelligence. *SSRN Electronic Journal*, 2023.
- [101] Kostas Karpouzis. Explainable ai for intelligent tutoring systems. In *Artificial Intelligence in Education*. Springer, 2024.
- [102] Qin Yangjie. Privacy-preserving federated learning framework in multimedia courses recommendation. *Wireless Networks*, 2022.
- [103] Bin Tian. A survey of personalized adaptive learning system. In *2023 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*. IEEE, 2023. Referencia compartida para personalización.
- [104] Bin Tian. A survey of personalized adaptive learning system. In *2023 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*. IEEE, 2023.
- [105] Ruth C Clark and Richard E Mayer. Scorm and emerging learning standards. *E-learning and the Science of Instruction*, pages 385–402, 2016.
- [106] Oleksandr Mosiiuk, Sergiy Kovalenko, and Andrey Petrov. xapi implementation in distributed microservices architectures. *Computers & Education*, 165, 2021.

- [107] Elena Lopez Martinez, Roberto Garcia, and Ana Silva. Mel: A four-layer adaptive learning architecture. *Educational Technology Research and Development*, 69, 2021.
- [108] Maheshya Weerasinghe, Aaron Quigley, Klen Čopić Pucihar, Alice Toniolo, Angela Miguel, and Matjaž Kljun. Arigatō: Effects of adaptive guidance on engagement and performance in augmented reality learning environments. In *ArXiv preprint*, 2022.
- [109] Shohida B. Yusupova, Sardorbek Zokirjon Ugli Davletboyev, and Bahrom Y. Ishmetov. Development of a flexible student training system using the moodle lms. In *2024 International Conference on Power, Information, Electronics and Robotics Engineering (PIERCE)*, pages 1–6. IEEE, 2024.
- [110] Li Wang, Xiaoli Chen, and Qiang Zhang. Restful interfaces for flexible lms integration. In *2020 International Conference on Educational Technology and Computer Science*, 2020.
- [111] Sarah Johnson, Michael Brown, and Jennifer Davis. Integrated analytics platforms for real-time learning insights. *Journal of Educational Computing Research*, 60, 2022.
- [112] Josep Prieto and David Gañán. A methodology approach to evaluate cloud-based infrastructures in support for e-assessment. In *Lecture Notes in Computer Science*. Springer, 2019.
- [113] K Amala, D Rajeswari, and Athish Venkatachalam Parthiban. Machine learning models for enhanced adaptive e-learning system design. In *Advances in Educational Technologies and Instructional Design*. IGI Global, 2025.
- [114] Asep Sutarman et al. A model-driven approach to developing scalable educational software for adaptive learning environments. *Information Technology in Education and Engineering*, 3(1):45–67, 2024.
- [115] Rianto Aradea and Husni Mubarak. Cultivating service knowledge models for iot-based systems adaptability. *Informatica*, 46, 2022.
- [116] Nabil Achtaich et al. A framework for smart iot adaptations. In *2020 International Conference on Smart Applications*, 2020.
- [117] Open edX Community. Open edx platform architecture documentation. <https://docs.openedx.org/en/latest/developers/concepts/architecture.html>, 2024. Consultado: 24 de enero de 2026.
- [118] Hitesh Choudhury. Feasibility study: Open-source e-learning platforms moodle vs efront. *Open Access Journal of Mathematical Research*, 1, 2025.
- [119] Bagus Jati Santoso et al. A docker container-based solution for course archival on moodle. 2023.

Bibliografía

- [120] Cindy Muhdiantini, Mokhammad Jamil, and Kertahadi Kertahadi. Optimizing learning environments: A literature review on microservices for moodle quality improvement. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 9(3):933–941, 2024.
- [121] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, Boston, MA, 2003.
- [122] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture: A System of Patterns*, volume 1. John Wiley & Sons, Chichester, UK, 1996.
- [123] Siamak Farshidi, Slinger Jansen, and Jan Martijn van der Werf. Capturing software architecture knowledge for pattern-driven design. *Journal of Systems and Software*, 170:110714, 2020.
- [124] Uwe Zdun et al. Capturing software architecture knowledge for pattern-based design. In *Proceedings of the 13th international conference on Software engineering*. Springer, 2008.
- [125] Messias Filho et al. Self-adaptive microservice-based systems. *arXiv preprint arXiv:2103.08688*, 2021.
- [126] João Figueira and Carlos Coutinho. Developing self-adaptive microservices. *Procedia Computer Science*, 232:264–273, 2024.
- [127] René F. Kizilcec and Hansol Lee. Algorithmic fairness in education. arXiv preprint arXiv:2007.05443, 2020. Preprint sobre equidad algorítmica en educación.
- [128] Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008. Retroalimentación formativa en educación.
- [129] John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77(1):81–112, 2007. El poder de la retroalimentación.
- [130] Advanced Distributed Learning (ADL). Scorm 2004 4th edition specification. Technical report, Advanced Distributed Learning Initiative, 2009.
- [131] Punya Mishra and Matthew J. Koehler. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6):1017–1054, 2006. Marco TPACK para conocimiento docente.
- [132] Matthew J. Koehler, Punya Mishra, and William Cain. What is technological pedagogical content knowledge (tpack)? *Journal of Education*, 193(3):13–19, 2013. Definición y aplicación del marco TPACK.
- [133] Abbas Fadhil Aljuboori and Hala Al-lawati. Intelligent adaptive e-learning systems: Current approaches, architectures, and applications. *European Conference on e-Learning*, 22(1):1925–1934, 2023.

- [134] Anthony Jameson. Adaptive interfaces and agents. In Julie A. Jacko and Andrew Sears, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 305–330. Lawrence Erlbaum Associates, Mahwah, NJ, 2003.

Bibliografía

Glosario

Accountability (Gobernanza y Rendición de Cuentas)

Capacidad de la arquitectura para garantizar la transparencia y trazabilidad de las decisiones algorítmicas, permitiendo auditar por qué se recomendó una ruta de aprendizaje específica.

ALS (Adaptive Learning System)

Sistema de aprendizaje capaz de modificar activamente su comportamiento, interfaz y contenidos en respuesta a las necesidades, preferencias y progreso detectado en el estudiante en tiempo real.

ARALS (Architecture of Reference for Adaptive Learning Systems)

Arquitectura de referencia propuesta en esta investigación que define el marco estructural, los componentes y las relaciones necesarias para transformar sistemas educativos tradicionales en ecosistemas adaptativos.

Artefacto Puente Interdisciplinario

Objeto técnico o conceptual, como ARALS, diseñado específicamente para facilitar la comunicación y la toma de decisiones compartidas entre pedagogos, arquitectos de software y desarrolladores.

C1 (Secuencia de Transformación Conceptual)

Fase inicial y crítica del proceso de evolución arquitectónica donde se define la transición lógica desde el modelo transaccional del LMS hacia el modelo adaptativo. Representa la inyección de inteligencia y la reconfiguración de componentes para superar la rigidez estructural (stiffness) del sistema legado.

Deuda Técnica Arquitectónica

Costo acumulado derivado de decisiones de diseño subóptimas en sistemas legados que dificultan la evolución hacia capacidades adaptativas sin una intervención estructural profunda.

LMS (Learning Management System)

Plataforma tecnológica diseñada para la gestión administrativa y transaccional de cursos, usuarios y contenidos educativos, caracterizada generalmente por una rigidez estructural frente a la personalización.

LRS (Learning Record Store)

Sistema de almacenamiento diseñado específicamente para gestionar y analizar trazas de aprendizaje enviadas mediante el estándar xAPI, actuando como la memoria histórica de la experiencia del estudiante.

LTI (Learning Tools Interoperability)

Estándar de la organización 1EdTech que facilita la integración transparente de herramientas de aprendizaje externas dentro de una plataforma anfitriona, garantizando la seguridad y la portabilidad de datos.

MAPE-K (Monitor-Analyze-Plan-Execute over Knowledge)

Modelo de control autónomo que rige la auto-adaptación mediante cuatro fases: monitoreo del entorno, análisis de datos, planificación de cambios y ejecución de ajustes basados en una base de conocimiento compartida.

Nexo PALS-A → PALS

Proceso metodológico de traducción técnica que vincula las capacidades estructurales y atributos de calidad (definidos como PALS-A en el Capítulo 3) con su instrumentación operativa y requerimientos funcionales (definidos como PALS en el Capítulo 5). Es el garante de la trazabilidad entre el diseño teórico de ARALS y la ejecución de la hoja de ruta.

PAC (Personalización Adaptativa Colaborativa)

Propiedad núcleo de la arquitectura ARALS que conceptualiza la adaptación como un proceso dinámico y continuo, integrando el perfil cognitivo del aprendiz con las dinámicas de colaboración de la comunidad de práctica.

PALS (Properties for Adaptive Learning Systems)

Instrumentación operativa e implementación funcional de las capacidades adaptativas en un sistema educativo. A diferencia de las PALS-A, que se centran en la estructura arquitectónica (Capítulo 3), las PALS representan la ejecución técnica, el despliegue de componentes de ARALS y el modelado de casos de uso para la transformación efectiva del sistema (Capítulo 5).

PALS-A (Properties for Adaptive Learning Systems - Architecturally influenceable)

Conjunto de capacidades y atributos de calidad de un sistema educativo cuya viabilidad técnica depende críticamente de las decisiones de diseño estructural, modularidad y niveles de acoplamiento.

Patrón Strangler Fig

Estrategia de refactorización de software que consiste en rodear un sistema monolítico legado con nuevas funcionalidades de manera incremental, hasta que el nuevo sistema reemplaza por completo la infraestructura original.

SoC (Separation of Concerns)

Principio de diseño que dicta que cada componente de la arquitectura debe ocuparse de una única funcionalidad delimitada, promoviendo el bajo acoplamiento y la alta cohesión.

xAPI (Experience API)

Estándar de interoperabilidad de e-learning que permite capturar datos de aprendizaje granulares en diversos contextos, utilizando sentencias de tipo actor-verbo-objeto para su almacenamiento en un LRS.