

FACULTAD DE INGENIERÍAS Y ARQUITECTURA
CARRERA PROFESIONAL DE INGENIERÍA DE SOFTWARE

Tesis Titulada:

**DESARROLLO DE UN SISTEMA INTEGRADO DE GESTIÓN
DOCUMENTAL Y ADMINISTRATIVA PARA DOCENTES DE LA
INSTITUCIÓN EDUCATIVA PADRE PÉREZ DE GUEREÑU DE LA
CIUDAD DE AREQUIPA**

Tesis presentada por los bachilleres:

**GUILLERMO JOSE ALEMAN ZAMBRANO
DANIEL GUSTAVO MENDIGURI CHAVEZ**

Asesor:

Dr. YASIEL PÉREZ VERA

Para optar por el título profesional de:

INGENIERO DE SOFTWARE

AREQUIPA – PERÚ

2025

DESARROLLO DE UN SISTEMA INTEGRADO DE GESTIÓN
DOCUMENTAL Y ADMINISTRATIVA PARA DOCENTES DE LA
INSTITUCIÓN EDUCATIVA PADRE PÉREZ DE GUEREÑU DE LA
CIUDAD DE AREQUIPA

INFORME DE ORIGINALIDAD



FUENTES PRIMARIAS

1	hdl.handle.net Fuente de Internet	1%
2	www.coursehero.com Fuente de Internet	1%
3	dspace.espoch.edu.ec Fuente de Internet	<1%
4	dspace.ups.edu.ec Fuente de Internet	<1%
5	es.slideshare.net Fuente de Internet	<1%
6	dspace.unitru.edu.pe Fuente de Internet	<1%
7	www.slideshare.net Fuente de Internet	<1%
8	repositorio.unap.edu.pe Fuente de Internet	<1%
9	Submitted to Universidad Cesar Vallejo Trabajo del estudiante	<1%

10	Submitted to Universidad TecMilenio Trabajo del estudiante	<1 %
11	repositorio.ulasalle.edu.pe Fuente de Internet	<1 %
12	repositorio.uns.edu.pe Fuente de Internet	<1 %
13	dspace.utalca.cl Fuente de Internet	<1 %
14	es.scribd.com Fuente de Internet	<1 %
15	www.scribd.com Fuente de Internet	<1 %
16	Submitted to Instituto Superior de Artes, Ciencias y Comunicación IACC Trabajo del estudiante	<1 %
17	community.devsense.com Fuente de Internet	<1 %
18	docentes20475lospelones.blogspot.com Fuente de Internet	<1 %
19	Submitted to Ilerna Online Blackboard Trabajo del estudiante	<1 %
20	Submitted to Arab Open University Trabajo del estudiante	<1 %
21	Submitted to University of Wales Swansea Trabajo del estudiante	<1 %
22	repositorio.upao.edu.pe Fuente de Internet	

		<1 %
23	fsegitlab.wlv.ac.uk Fuente de Internet	<1 %
24	www.dropbox.com Fuente de Internet	<1 %
25	alicia.concytec.gob.pe Fuente de Internet	<1 %
26	code.tutsplus.com Fuente de Internet	<1 %
27	Submitted to Universidad Tecnológica Centroamericana UNITEC Trabajo del estudiante	<1 %
28	Submitted to University of Northampton Trabajo del estudiante	<1 %
29	doczz.es Fuente de Internet	<1 %
30	pdfcookie.com Fuente de Internet	<1 %
31	translate.evernote.com Fuente de Internet	<1 %
32	Submitted to Chester College of Higher Education Trabajo del estudiante	<1 %
33	repositorio.ucv.edu.pe Fuente de Internet	<1 %
	www.espacio.digital.upel.edu.ve	

34	Fuente de Internet	<1 %
35	core.ac.uk Fuente de Internet	<1 %
36	repositorio.uladech.edu.pe Fuente de Internet	<1 %
37	repositoriocyt.unlam.edu.ar Fuente de Internet	<1 %
38	vdocumento.com Fuente de Internet	<1 %
39	repositorio.utp.edu.co Fuente de Internet	<1 %
40	Submitted to UNIBA Trabajo del estudiante	<1 %
41	Submitted to Universidad Nacional de San Cristóbal de Huamanga Trabajo del estudiante	<1 %
42	docentesmariateguistas.blogspot.com Fuente de Internet	<1 %
43	oldri.ues.edu.sv Fuente de Internet	<1 %
44	repositorio.unfv.edu.pe:8080 Fuente de Internet	<1 %
45	opac.pucv.cl Fuente de Internet	<1 %
46	repositorio.upn.edu.pe Fuente de Internet	<1 %

47	Gomez Huacso, Alexander Saul. "Analisis de una Praxeologia Matematica de las Inecuaciones Lineales en los Libros Didacticos de Educacion Secundaria.", Pontificia Universidad Catolica del Peru - CENTRUM Catolica (Peru), 2020	<1 %
48	Submitted to Universidad Abierta para Adultos	<1 %
49	Submitted to Universidad Internacional del Ecuador	<1 %
50	Submitted to Universidad Tecnologica del Peru	<1 %
51	es.digitaltrends.com	<1 %
52	saber.ucv.ve	<1 %
53	www.rclick.co.uk	<1 %
54	Submitted to Corporación Universitaria Minuto de Dios,UNIMINUTO	<1 %
55	Submitted to Pontificia Universidad Catolica del Ecuador - PUCE	<1 %

56	Submitted to Universidad APEC Trabajo del estudiante	<1 %
57	Submitted to Universidad Andrés Bello Trabajo del estudiante	<1 %
58	Submitted to Universidad de Guayaquil Trabajo del estudiante	<1 %
59	Submitted to University of Hertfordshire Trabajo del estudiante	<1 %
60	keepcoding.io Fuente de Internet	<1 %
61	e-archivo.uc3m.es Fuente de Internet	<1 %
62	Submitted to ipn Trabajo del estudiante	<1 %
63	repository.upb.edu.co:8080 Fuente de Internet	<1 %
64	vdocuments.es Fuente de Internet	<1 %
65	Huamán, Segundo Oswaldo Pastor. "Efectividad del uso de la Plataforma Google Classroom para Potenciar el Desarrollo de Competencias Digitales en Estudiantes del 5to de Secundaria de una Institución Educativa Privada de Lima Metropolitana.", Pontificia Universidad Católica del Perú (Peru) Publicación	<1 %

- 66 Mendoza, Claudia Maria Ricardina Atencio. "La inteligencia emocional y el desempeño docente en la Escuela Profesional de Arte de la UNA - Puno, 2019 - I", Universidad Nacional del Altiplano de Puno (Peru) <1 %
Publicación
-
- 67 Velázquez Román, Luis E.. "Using eBird to Assess Avian Resistance and Resilience After Hurricanes Irma and María in the San Juan Metropolitan Area", University of Puerto Rico, Rio Piedras (Puerto Rico) <1 %
Publicación
-
- 68 bibliographie-trac.ub.rub.de <1 %
Fuente de Internet
-
- 69 jack30.com <1 %
Fuente de Internet
-
- 70 joshuaturner0.blogspot.com <1 %
Fuente de Internet
-
- 71 metodologiasagiles.org <1 %
Fuente de Internet
-
- 72 social.technet.microsoft.com <1 %
Fuente de Internet
-
- 73 www.clubensayos.com <1 %
Fuente de Internet
-
- 74 www.henkel-northamerica.com <1 %
Fuente de Internet
-
- 75 "Applied Technologies", Springer Science and Business Media LLC, 2023 <1 %

Publicación

76	ftpc.borland.com Fuente de Internet	<1 %
77	github.com Fuente de Internet	<1 %
78	grove.ccsd59.org Fuente de Internet	<1 %
79	ideba.gba.gob.ar Fuente de Internet	<1 %
80	oa.upm.es Fuente de Internet	<1 %
81	repositorio.uch.edu.pe Fuente de Internet	<1 %
82	repositorio.utc.edu.ec Fuente de Internet	<1 %
83	repositorioacademico.upc.edu.pe Fuente de Internet	<1 %
84	tesis.pucp.edu.pe Fuente de Internet	<1 %
85	upc.aws.openrepository.com Fuente de Internet	<1 %
86	www.aniei.org.mx Fuente de Internet	<1 %
87	www.grade.org.pe Fuente de Internet	<1 %
88	www.grupocatalanaoccidente.com Fuente de Internet	

		<1 %
89	www.panoramaaudiovisual.com Fuente de Internet	<1 %
90	www.pmesut.gob.pe Fuente de Internet	<1 %
91	www.researchgate.net Fuente de Internet	<1 %
92	www.topocal.com Fuente de Internet	<1 %
93	1library.co Fuente de Internet	<1 %
94	Submitted to De Montfort University Trabajo del estudiante	<1 %
95	Pedro López Gómez. "Archival science in Spain between 1975 and 2005: a review", <i>Archival Science</i> , 2008 Publicación	<1 %
96	Silvia Marzal Romeu. "Concepción e integración de arquitecturas y protocolos de comunicación dentro de sistemas de supervisión y control de microrredes inteligentes", <i>Universitat Politecnica de Valencia</i> , 2019 Publicación	<1 %
97	Submitted to University of Cincinnati Trabajo del estudiante	<1 %
	bibdigital.epn.edu.ec	

98	Fuente de Internet	<1 %
99	candy8.net Fuente de Internet	<1 %
100	documentos.una.ac.cr Fuente de Internet	<1 %
101	dspace-uh-tmp.igniteonline.la Fuente de Internet	<1 %
102	dspace.usc.es Fuente de Internet	<1 %
103	edicioncero.cl Fuente de Internet	<1 %
104	elvex.ugr.es Fuente de Internet	<1 %
105	es-knowledge.faro.com Fuente de Internet	<1 %
106	es.imaginecommunications.com Fuente de Internet	<1 %
107	esdocs.com Fuente de Internet	<1 %
108	fr.slideshare.net Fuente de Internet	<1 %
109	funredes.org Fuente de Internet	<1 %
110	gitlab.mitic.gov.py Fuente de Internet	<1 %

111	myslide.es Fuente de Internet	<1 %
112	patents.google.com Fuente de Internet	<1 %
113	prezi.com Fuente de Internet	<1 %
114	repositorio.uis.edu.co Fuente de Internet	<1 %
115	tesis.ucsm.edu.pe Fuente de Internet	<1 %
116	utnba.centrodelearning.com Fuente de Internet	<1 %
117	www.exact.com.pe Fuente de Internet	<1 %
118	www.sartorius.com Fuente de Internet	<1 %
119	Sufyan bin Uzayr. "Bootstrap - The Ultimate Guide", CRC Press, 2022 Publicación	<1 %

Excluir citas

Activo

Excluir coincidencias

Apagado

Excluir bibliografía

Activo

DEDICATORIA

A nuestras familias, quienes con su amor incondicional, sacrificio y apoyo constante han sido nuestro pilar fundamental en cada paso de este camino. Gracias por creer en nosotros, por impulsarnos a perseguir nuestros sueños y por estar presentes en cada momento, en las alegrías y en los desafíos. Este logro es el reflejo de su esfuerzo y dedicación. Se los dedicamos con todo nuestro amor y gratitud

AGRADECIMIENTOS

La culminación de este trabajo de tesis no habría sido posible sin el apoyo invaluable de muchas personas a quienes deseamos expresar nuestro más profundo agradecimiento.

A nuestro tutor, Yasiel, por su guía constante, paciencia y dedicación a lo largo de este proceso. Sus conocimientos, tiempo y apoyo fueron fundamentales para el desarrollo y culminación de esta investigación. Gracias por creer en nosotros y por impulsarnos a dar siempre lo mejor.

A nuestros padres, madres y hermanos, quienes han sido nuestro pilar fundamental en cada etapa de nuestras vidas. Su amor incondicional, sacrificio y apoyo en todo momento nos han dado la fortaleza necesaria para alcanzar nuestras metas. Esta tesis es también un logro de ustedes.

A nuestros amigos de la universidad, quienes compartieron con nosotros no solo conocimientos, sino también risas, desafíos y momentos inolvidables durante toda la carrera. Su compañía hizo este camino más llevadero y significativo.

A todos ustedes, gracias por formar parte de este logro.

Índice general

Dedicatoria	I
Agradecimientos	II
Índice de Abreviaturas y Siglas	1
Índice de Tablas	2
Índice de Figuras	4
Glosario de Términos	5
Resumen	8
Abstract	9
Palabras Clave	10
I Problemática del Proyecto	11
1.1 Contexto del Problema	11
1.2 Antecedentes y Estado del Arte	12
1.2.1 Antecedentes Nacionales	12
1.2.2 Antecedentes Internacionales	13
1.3 Definición del Problema	14
1.3.1 Problemática Identificada	14
1.3.2 Características del Problema	15
II Planteamiento del Proyecto	16
2.1 Fundamentos Teóricos	16
2.1.1 Ingeniería de software	16
2.1.2 Ingeniería de requerimientos	16

2.1.3	GWT	17
2.1.4	Metodologías de Desarrollo de Software Ágil	17
2.1.5	Sistemas de información	18
2.1.6	Framework Laravel	19
2.1.7	PostgreSQL	20
2.1.8	Bootstrap	21
2.1.9	HTTP Apache	21
2.1.10	Arquitectura de Software (MVC - Modelo Vista Controlador).....	21
2.1.11	Patrones de Diseño	22
2.1.12	Estilo arquitectónico REST	22
2.1.13	API REST	23
2.1.14	Seguridad en Aplicaciones Web	23
2.1.15	Sistemas de Gestión Documental (DMS).....	23
2.1.16	Sistemas de Gestión Documental en Instituciones educativas	24
2.1.17	Marco Curricular Nacional de la Educación Básica del Perú	24
2.1.18	Planificación curricular y de áreas	24
2.1.19	Documentos académicos de la institución educativa Padre Pérez De Guereña	26
2.2	Objetivos del Proyecto	27
2.2.1	Objetivo General	27
2.2.2	Objetivos Específicos	27
2.3	Justificación	27
2.4	Viabilidad	29
2.5	Limitaciones	30
III	Metodología de Desarrollo	32
3.1	Plan de Gestión del Proyecto.....	32
3.1.1	Alcance	32
3.1.2	Metodología de desarrollo	34
3.1.3	Planificación e historias de usuario	37
3.1.4	Recursos del Proyecto	41
3.2	Ánalisis y Diseño del Sistema.....	43
3.2.1	Requerimientos del Sistema	43
3.2.2	Casos de Uso	44
3.2.3	Requerimientos No Funcionales	56
3.2.4	Suposiciones y Restricciones	56

3.2.5	Patrones de Diseño Implementados	58
3.2.6	Consideraciones de Integridad y Consistencia	59
3.3	Diagrama de Secuencia UML	60
3.4	Arquitectura del Sistema.....	63
3.5	Diseño de la Base de Datos	65
3.5.1	Arquitectura de Base de Datos	65
3.5.2	Normalización	67
3.6	Implementación del Sistema	68
3.6.1	Tecnologías Utilizadas	68
3.6.2	Desarrollo de los Módulos	70
3.6.3	Integración del Sistema	80
3.6.4	Estrategia de Pruebas	81
3.6.5	Pruebas Funcionales	84
3.6.6	Pruebas de Usabilidad	85
IV	Resultados y Discusión	86
4.1	Descripción de la Situación Actual	86
4.1.1	Contexto Institucional	86
4.1.2	Problemáticas Detectadas	86
4.1.3	Métodos de Trabajo Previos	88
4.2	Implementación del Software Desarrollado	88
4.2.1	Proceso de Instalación y Configuración	88
4.2.2	Capacitación Inicial a los Docentes	89
4.2.3	Desarrollo bajo Metodología Scrum	89
4.2.4	Iteraciones de Validación	90
4.2.5	Importancia de las historias de usuario en el desarrollo ágil	91
4.3	Validación del Software Desarrollado	92
4.3.1	Proceso de Validación Continua	92
4.3.2	Formulario de Satisfacción	92
4.3.3	Pruebas de Usabilidad	94
4.4	Métricas y Análisis Comparativo (Antes y Despues)	94
4.4.1	Tiempo Promedio Semanal en Tareas Administrativas	94
4.4.2	Pérdida de Documentos	95
V	Conclusiones	97
Bibliografía		100

Anexos	104
Diagrama de Gantt.....	104
Formulario de Satisfacción	105

ÍNDICE DE ABREVIATURAS Y SIGLAS

Abreviatura/Sigla	Significado
AWS	Amazon Web Services
CSS	Cascading Style Sheets
DMS	Document Management System (Sistema de Gestión Documental)
HTML	HyperText Markup Language
IE	Institución Educativa
JS	JavaScript
MCN	Marco Curricular Nacional
MINEDU	Ministerio de Educación del Perú
PHP	PHP: Hypertext Preprocessor
PostgreSQL	Sistema de gestión de bases de datos relacional
Scrum	Framework de gestión de proyectos ágiles
SQL	Structured Query Language
TICS	Tecnologías de la Información y Comunicación
UI	User Interface (Interfaz de Usuario)
ACL	Listas de Control de Acceso
ACID	Atomicidad, Consistencia, Aislamiento, Durabilidad

Índice de Tablas

1	Desglose de Costos y Ahorros	30
2	Historias de Usuario y Componentes del Sistema (Parte 1)	37
3	Historias de Usuario y Componentes del Sistema (Parte 2)	38
4	Criterios de Aceptación (Given-When-Then)	38
5	Recursos del Proyecto (Parte 1)	41
6	Recursos del Proyecto (Parte 2)	42
7	Requerimientos Funcionales (Parte 1)	43
8	Requerimientos Funcionales (Parte 2)	44
9	Stack Tecnológico del Sistema	70
10	Resumen de pruebas funcionales ejecutadas	84
11	Resultados cuantitativos de las pruebas de usabilidad	85
12	Resultados cuantitativos del formulario de satisfacción	92
13	Comparación de tiempo semanal dedicado a tareas administrativas	94
14	Comparación de incidentes de pérdida documental	95

Índice de figuras

1	Metodología Scrum según [32]	34
2	Caso de uso RF-01: Inicio de sesión	45
3	Caso de uso RF-01: Registrar usuario	45
4	Caso de uso RF-02: Recuperación de contraseña	45
5	Caso de uso RF-03: Roles y permisos	46
6	Caso de uso RF-04: Subida de documentos	47
7	Caso de uso RF-05: Organización de documentos	47
8	Caso de uso RF-06: Edición de perfil de docente	48
9	Caso de uso RF-07: Control de versiones de documentos	48
10	Caso de uso RF-08: Restauración de versiones de documentos	49
11	Caso de uso RF-09: Búsqueda de documentos	49
12	Caso de uso RF-10: Etiquetado de documentos	49
13	Caso de uso RF-11: Descarga segura de documentos	50
14	Caso de uso RF-12: Eliminación lógica de documentos	50
15	Caso de uso RF-13: Administración de áreas, grados y secciones	51
16	Caso de uso RF-14: Administración de perfiles de docentes	51
17	Caso de uso RF-15: Visualización de calendario responsive	52
18	Caso de uso RF-16: Visualización de calendario según rol	52
19	Caso de uso RF-17: Creación y edición manual de horarios con Undo/Redo ..	53
20	Caso de uso RF-18: Detección de conflictos en horarios	53
21	Caso de uso RF-19: Asignación automática de docentes a áreas	54
22	Caso de uso RF-20: Aceptación y ajuste manual de asignaciones automáticas ..	54
23	Caso de uso RF-21: Exportación de horarios a PDF y CSV	55
24	Caso de uso RF-22: Backup automático mensual	55
25	Caso de uso RF-23: Control de accesos mediante ACL	56
26	Modelo de Clases UML	57
27	Diagrama de secuencia DS-03: Detección de Conflictos en Horarios	60
28	Diagrama de secuencia DS-04: Detección de Conflictos en Horarios	61
29	Diagrama de secuencia DS-01: Asignación Automática de Docentes	62

30	Diagrama de secuencia DS-02: Detección de Conflictos en Horarios	63
31	Diagrama de la arquitectura del sistema	64
32	Diagrama de entidad–relación de la base de datos	66
33	Imagen tomada del login del sistema web.....	76
34	Imagen tomada de la vista de inicio del sistema web logueado como administrador.....	77
35	Comparación del tiempo administrativo promedio por docente antes y después de la implementación.	95
36	Porcentaje de docentes que informaron pérdida de documentos antes y después de usar el sistema.....	96
37	Tiempo de planificación del coordinador: 35 horas (rango 30-40) antes vs. 8 horas después del sistema.	96

GLOSARIO DE TÉRMINOS

Algoritmo de asignación automática	Conjunto de reglas y procedimientos lógicos, de autoría propia, diseñados para optimizar la distribución de docentes en áreas y horarios, considerando restricciones de disponibilidad, carga laboral y áreas de conocimiento.
Arquitectura de información	Estructura y diseño organizacional de los datos y contenidos dentro del sistema, modelada en torno a los conceptos clave del Marco Curricular Nacional (competencias, capacidades, desempeños, estándares) para alinearse con el flujo de trabajo docente.
Backup	Proceso de crear y almacenar una copia de los datos del sistema para permitir su recuperación en caso de una pérdida de información original debido a fallos de hardware, software o desastres.
Carga administrativa	Conjunto de tareas y procedimientos de naturaleza no pedagógica (como la gestión de documentos, formularios y planificación horaria) que los docentes deben realizar, consumiendo tiempo que podría dedicarse a labores educativas directas.
Digitalización de procesos	Conversión de procedimientos manuales o basados en papel en flujos de trabajo electrónicos y automatizados, con el objetivo de aumentar la eficiencia, reducir errores y mejorar la accesibilidad de la información.
Diversificación Curricular	Proceso de adaptación del currículo nacional a las características específicas de los estudiantes y el contexto de la institución educativa, definiendo competencias, capacidades, desempeños y campos temáticos para cada grado y área.

Extrapapelamiento	Término coloquial utilizado en el contexto educativo para describir la problemática de la pérdida o mala gestión de documentos físicos.
Framework Laravel	Herramienta de desarrollo web de código abierto basada en PHP, utilizada para construir la aplicación propuesta. Facilita un desarrollo estructurado, seguro y escalable mediante el uso de patrones de diseño y componentes preconstruidos.
Gestión documental	Conjunto de prácticas y herramientas para administrar el ciclo de vida de los documentos electrónicos, incluyendo su creación, almacenamiento, clasificación, recuperación, distribución y eliminación controlada.
Interfaz responsiva	Diseño de la interfaz de usuario (UI) que se adapta automáticamente al tamaño y orientación de la pantalla del dispositivo desde el cual se accede (computadora, tableta, teléfono móvil), garantizando una buena experiencia de usuario.
Marco Curricular Nacional (MCN)	Instrumento de política educativa peruana que establece los aprendizajes fundamentales que deben lograr los estudiantes. Define los conceptos estructurantes de competencias, capacidades, estándares de aprendizaje y desempeños.
Metodología Scrum	Marco de trabajo ágil para la gestión y desarrollo de proyectos software. Se caracteriza por su iteratividad (Sprints), roles definidos (Product Owner, Scrum Master, Equipo de Desarrollo) y eventos regulares (Planificación, Daily, Revisión, Retrospectiva) para fomentar la adaptación y la mejora continua.
Soft-delete	Estrategia de diseño donde un registro no se borra físicamente de la base de datos, sino que se marca con un flag o estado como “eliminado”. Permite recuperar datos accidentalmente borrados y mantener la integridad referencial histórica.
Stack tecnológico	Conjunto de tecnologías, lenguajes de programación, frameworks y herramientas utilizadas para el desarrollo de una aplicación software. En este proyecto: Laravel (PHP), PostgreSQL, JavaScript, HTML, CSS.

Usabilidad

Atributo de calidad del software que evalúa la facilidad con la que los usuarios (docentes y administrativos) pueden aprender a usar el sistema, eficiencia para realizar tareas, facilidad para recordar su funcionamiento y satisfacción subjetiva con su uso.

RESUMEN

El presente trabajo tuvo como objetivo diseñar e implementar un sistema integrado de gestión documental y administrativa dirigido a los docentes de la Institución Educativa Padre Pérez de Guereña de la ciudad de Arequipa. La investigación abordó la problemática de la sobrecarga administrativa y la gestión manual de documentos académicos, que generan pérdida de tiempo, duplicidad de información y estrés laboral en el personal docente. Para ello, se aplicó la metodología ágil Scrum, desarrollando un software web basado en tecnologías de código abierto como Laravel, PostgreSQL y Bootstrap. El sistema se estructuró en módulos académico y administrativo, permitiendo la organización, almacenamiento y recuperación eficiente de los documentos institucionales.

Durante la fase de validación, se ejecutaron pruebas funcionales, de integración y de usabilidad, a los docentes (usuarios finales) se les presentó un formulario de Google para tener conocimiento de su aceptación, estos reflejaron una buena aceptación general y áreas potenciales de mejora en la experiencia de usuario. Los resultados demostraron que la solución propuesta reduce significativamente la carga administrativa docente y optimiza los procesos de gestión documental. En conclusión, el sistema constituye una herramienta efectiva y escalable que contribuye a la digitalización educativa y mejora de la eficiencia institucional.

ABSTRACT

The present work had as objective to design and implement an integrated system of documental and administrative management aimed at the teachers of the Padre Pérez de Guereñu Educational Institution in the city of Arequipa. The research addressed the problematic of the administrative overload and the manual management of academic documents, which generate loss of time, duplication of information, and work stress in the teaching staff. For this, the agile methodology Scrum was applied, developing a web software based on open-source technologies such as Laravel, PostgreSQL, and Bootstrap. The system was structured into academic and administrative modules, allowing the organization, storage, and efficient retrieval of institutional documents.

During the validation phase, functional, integration, and usability tests were executed; a Google form was presented to the teachers (end-users) to obtain knowledge of their acceptance, these reflected a good general acceptance and potential areas for improvement in the user experience. The results demonstrated that the proposed solution significantly reduces the teaching administrative burden and optimizes the documental management processes. In conclusion, the system constitutes an effective and scalable tool that contributes to educational digitalization and the improvement of institutional efficiency.

PALABRAS CLAVE

Gestión documental, Carga administrativa docente, Software educativo, Metodologías ágiles, Sistema web.

Capítulo I - PROBLEMÁTICA DEL PROYECTO

1.1. Contexto del Problema

En los últimos años, el sistema educativo peruano ha experimentado transformaciones significativas orientadas hacia la mejora de la calidad educativa y la modernización de sus procesos institucionales. Sin embargo, los docentes de las diferentes instituciones educativas continúan enfrentando una carga administrativa considerable que se ha convertido en un obstáculo para el ejercicio de su función pedagógica. Esta sobrecarga deriva principalmente de la gestión manual de documentos académicos, la coordinación de recursos pedagógicos y la organización de horarios, procesos que demandan tiempo y esfuerzo que podrían destinarse a la preparación de clases, el diseño de estrategias didácticas y la atención personalizada a los estudiantes.

La problemática se incrementa cuando estos procesos se llevan a cabo de manera manual o mediante herramientas digitales genéricas como Google Classroom [1], las cuales, si bien facilitan ciertos aspectos de la comunicación educativa, no están diseñadas para alinearse con las particularidades del Marco Curricular Nacional (MCN) peruano ni con las necesidades administrativas específicas de las instituciones educativas locales. Estas herramientas carecen de funcionalidades especializadas para la gestión de competencias, capacidades, desempeños y estándares de aprendizaje definidos por el Ministerio de Educación (MINEDU), lo que obliga a los docentes a realizar procesos adicionales de adaptación, reformateo y duplicación de información.

Además, la ausencia de sistemas centralizados de gestión documental provoca problemas recurrentes como la pérdida de archivos, la generación de múltiples versiones de un mismo documento sin control de cambios, y la dificultad para recuperar información histórica necesaria para la planificación anual. Por otro lado, la asignación de horarios y la distribución de carga académica, procesos que tradicionalmente se realizan de forma manual por los coor-

dinadores académicos, generan conflictos de disponibilidad, solapamientos y distribuciones erróneas que impactan negativamente en la organización institucional y en el bienestar docente.

El contexto actual exige soluciones tecnológicas contextualizadas que no solo digitalicen los procesos administrativos, sino que además se alineen con las políticas educativas nacionales, respeten los flujos de trabajo institucionales y promuevan la eficiencia operativa sin añadir complejidad innecesaria. La transformación digital en educación, reconocida como una prioridad nacional [2], representa una oportunidad para diseñar sistemas integrales que respondan a las necesidades reales de los docentes y contribuyan a mejorar la calidad del servicio educativo en el Perú.

1.2. Antecedentes y Estado del Arte

1.2.1. Antecedentes Nacionales

El área de estudio más importante para conseguir una mejora eficiente en las instituciones educativas del Perú es la digitalización de los procesos de gestión académica. La literatura académica muestra en evidencia la problemática derivada de los procesos manuales y la necesidad de implementar soluciones tecnológicas en el flujo de trabajo.

En [3] se propuso un sistema de gestión de sílabos para la Universidad Católica de Santa María. La investigación tuvo como problemática principal la gestión manual de sílabos, este inconveniente generaba errores, retrasos e inconsistencias. La solución propuesta en [3] fue una aplicación web desarrollada con .NET Core y React; esta aplicación web demostró un gran impacto, se proyectó que se haría una reducción del 60 % en el tiempo de elaboración de los sílabos juntos con un ahorro de 153.060 soles anuales estimados. En este estudio pudimos observar que se validó la necesidad de digitalizar la gestión documental pudiendo estandarizar los documentos y se mostró una mayor eficiencia reduciendo la carga de trabajo. Asimismo encontramos que el estudio se limita exclusivamente a los sílabos, mientras que nuestra propuesta abarca toda la planeación académica desde la diversificación hasta los recursos de cada sesión.

Siguiendo con las Instituciones de educación superior, en el trabajo [4] se presenta la implementación de una “Carpeta Pedagógica” web que mejora la gestión académica. La im-

plementación de este sistema surge debido a la necesidad de reducir los procesos manuales y la demora en la gestión de documentos de la Universidad Nacional del Santa. Estos procesos manuales afectaban directamente al cronograma de actividades teóricas, el plan de sesiones de clases por semanas, registro de asignaturas y registros de sílabos junto con su matriz de evaluación.

Estos inconvenientes generaban en los docentes una constante insatisfacción, debido a que consideraban que el tiempo invertido podría ser usado para tareas más relevantes. La carpeta pedagógica se desarrolló con la metodología OOHDm utilizando Java Spring Boot. Según se indica en [4], la satisfacción de los docentes aumentó del 17.4 % al 78.26 % después de su implementación.

Este estudio muestra cómo la correcta organización de los documentos y planificación de sesiones impacta directamente en la satisfacción y eficiencia de los docentes. Nuestra propuesta comparte el mismo objetivo, con la diferencia de que agregaremos la funcionalidad de administración de horarios.

En [5] se implementó en la institución educativa Juan Pablo II un sistema web para optimizar el registro de notas y matrículas. Para la arquitectura de dicho sistema se utilizó .NET y SQL Server, junto con una metodología Scrum, logrando así una reducción del 60 % en los tiempos de los procesos, junto con una aceptación del 100 % por parte de los docentes. En [5] se nos muestra la viabilidad de implementar un sistema web en los colegios peruanos junto con la alineación con las normativas del MINEDU.

[6] nos muestra un estudio realizado en el instituto técnico Edutech en Trujillo. Este estudio nos muestra una vez más que el uso de Laravel/PHP es un elemento factible para nuestro stack tecnológico, [6] optimizó el acceso a enlaces de clases y envío de tareas mostrándonos una vez más la importancia de concebir sistemas integrales en las instituciones educativas independientemente del nivel educativo.

1.2.2. Antecedentes Internacionales

Si bien nuestro estudio se encuentra en el contexto del peruano vimos por necesario mostrar los estudios de Ecuador complementando el panorama con sus problemáticas y soluciones convergentes.

[7] nos muestra cómo se implementó en la Unidad Educativa Hermano Miguel de la ciu-

dad de Latacunga, Ecuador una aplicación web de gestión académica, la implementación de dicho sistema surgió como solución a la gestión manual de asignación de docentes, inscripciones y matrículas entre otras; esta administración manual era propensa a pérdida de archivos físicos y errores debido a la alta carga de trabajo del personal, el sistema web implementado en [7] logró reducir el tiempo y carga de trabajo, los resultados obtenidos demostraron que el 91 % de docentes y secretaría afirmó que la aplicación web evitará la pérdida de información. Las limitaciones que presenta el trabajo es que muchos de los integrantes de la comunidad de la institución no cuentan con conocimientos de las TICS, es por ellos que se debieron dar capacitaciones las cuales dificultaron la adopción y uso efectivo del sistema web, también se indica que el mantenimiento continuo del sistema web es esencial para mantener el sistema funcional, aunque se nos aclara que esto no es debido al diseño del software sino a una consideración operativa para cerciorarse del correcto funcionamiento.

Dentro de la literatura encontramos dos trabajos de relevancia [8] y [9], estos trabajos fueron realizados en la Escuela Politécnica Nacional de Ecuador, estos trabajos complementarios buscaron reemplazar el uso del software FET el cual era usado para la planificación de horarios, el primero se centró en crear una aplicación web la cual gestiona las restricciones de disponibilidad de los docentes para poder coordinar sus horarios, mientras que el segundo se centró en la gestión de registros académicos. Ambos trabajaron sus problemáticas de forma modular y separada. Lo que nuestra propuesta busca es poder integrar tanto las capacidades de restricciones y recursos dentro de un único sistema el cual pueda crear un ambiente idóneo para los docentes y su administración.

1.3. Definición del Problema

1.3.1. Problemática Identificada

Entre las principales dificultades identificadas se encuentran:

- **Pérdida de documentos:** Comúnmente llamada “extrapapelamiento”, esta problemática surge debido a la falta de sistemas centralizados que permitan un almacenamiento seguro y accesible de la información.
- **Duplicidad de esfuerzos:** Los docentes deben elaborar repetidamente documentos con formatos similares (diversificaciones curriculares, planificaciones anuales, sesiones de aprendizaje) año tras año, debido a la falta de plantillas estandarizadas y mecanismos de reutilización.

zación de información. Esto implica un gasto innecesario de tiempo y energía en tareas administrativas repetitivas que podrían ser automatizadas o simplificadas mediante sistemas adecuados. La ausencia de versionado de documentos dificulta la recuperación de información histórica, obligando a los docentes a rehacer desde cero documentos que ya habían sido elaborados en períodos anteriores.

- **Errores en la asignación de horarios y recursos:** La planificación de horarios se realiza de forma manual, sin herramientas automatizadas que validen restricciones como disponibilidad docente, solapamiento de clases, distribución equitativa de carga académica y competencias específicas por área. Esta ausencia de validación automática genera conflictos recurrentes, asignaciones inapropiadas y desbalances en la carga de trabajo, afectando tanto la organización escolar como el bienestar de los docentes.
- **Sobrecarga de procesos administrativos en los docentes:** Los docentes enfrentan una carga administrativa considerable la cual deriva de la gestión manual de documentos académicos como planificaciones y registros de evaluación.

1.3.2. Características del Problema

El problema identificado cumple con las siguientes condiciones que justifican el esfuerzo de investigación:

- **Originalidad:** Si bien existen sistemas de gestión educativa, la propuesta integra gestión documental con planificación de horarios específicamente adaptada al Marco Curricular Nacional peruano y a las necesidades de la IE Padre Pérez de Guereñu.
- **Trascendencia:** La solución impactará directamente en la calidad educativa al liberar tiempo de los docentes para actividades pedagógicas y reducir el estrés administrativo.
- **Actualidad:** La transformación digital es una prioridad para el sector educativo peruano según [2].
- **Relevancia:** El 84.4 % de los docentes peruanos reportan sentir estrés debido a la gestión administrativa [10].
- **Aplicación de conocimientos:** El proyecto constituye una oportunidad integral para aplicar los conocimientos adquiridos durante la carrera de Ingeniería de Software, abarcando áreas fundamentales como ingeniería de requisitos, diseño de arquitecturas de software, modelado de datos, desarrollo web full-stack, gestión ágil de proyectos, pruebas de software y evaluación de usabilidad. El proyecto permite además desarrollar competencias transversales como trabajo en equipo, comunicación con stakeholders, resolución de problemas complejos y pensamiento crítico aplicado a contextos reales.

Capítulo II - PLANTEAMIENTO DEL PROYECTO

2.1. Fundamentos Teóricos

2.1.1. Ingeniería de software

Según [11], la ingeniería de software se define como: “the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates”. Esta definición puede traducirse como: “la disciplina que se ocupa de la producción y el mantenimiento de productos de software que se desarrollan y modifican a tiempo y dentro del presupuesto estimado”.

La adopción de este concepto garantiza un trabajo estructurado que optimiza los recursos y la gestión de la complejidad del desarrollo, permitiendo asegurar un producto de software de calidad, y no el resultado de una serie de esfuerzos improvisados.

2.1.2. Ingeniería de requerimientos

En [12] se define la Ingeniería de Requisitos como: “La Ingeniería de Requisitos abarca un conjunto de actividades organizadas para identificar y especificar lo que un sistema debe hacer, y asegurarse de que esté alineado con los objetivos del negocio y las restricciones del entorno”.

Adoptar estos conocimientos y fases para la toma de requerimientos es esencial para nuestro proyecto, ya que nos permite contar con un marco de referencia para su correcta ejecución, asegurando que los requerimientos no solo estén bien documentados, sino que también respondan a las necesidades de la Institución Educativa Padre Pérez de Guereñu. De esta manera, se minimiza el riesgo de desarrollar funcionalidades innecesarias que podrían generar

una gestión ineficiente del tiempo y extender la fase de desarrollo.

2.1.3. GWT

El paradigma Given–When–Then (GWT) es una estructura lingüística fundamental y obligatoria asociada principalmente al lenguaje Gherkin. Gherkin es la herramienta clave en el Desarrollo Guiado por Comportamiento (BDD) para definir el comportamiento esperado de un sistema. La popularidad del lenguaje Gherkin reside en la simplicidad de su estructura GWT, combinada con su capacidad para la verificación. Un requisito GWT consiste en una tripleta $\langle G, W, T \rangle$ que describe escenarios de comportamiento de manera clara y no ambigua. Las tres palabras clave son fundamentales:

- **Given (Dado):** Describe el contexto inicial del escenario. Debe establecer el sistema en un estado conocido antes de cualquier interacción.
- **When (Cuando):** Representa un evento o una acción. Esto puede ser una interacción del usuario o la recepción de un mensaje de otro sistema.
- **Then (Entonces):** Expresa el estado final deseado del sistema después de que el evento de When ha ocurrido.

Esta estructura se utiliza para capturar requisitos como relaciones de causa–efecto, sirviendo los escenarios Gherkin como especificaciones ejecutables para las pruebas de aceptación [13].

2.1.4. Metodologías de Desarrollo de Software Ágil

Según [14], las metodologías ágiles constituyen una alternativa sustancial a los enfoques metodológicos tradicionales en el desarrollo de software. El surgimiento de estas metodologías responde principalmente a la necesidad de mitigar los elevados índices de fracaso en proyectos de desarrollo de software. La implementación de marcos de trabajo ágiles permite reducir significativamente la carga administrativa y burocrática inherente a las metodologías convencionales.

Las características distintivas de estos enfoques metodológicos se fundamentan en dos principios centrales: la flexibilidad operativa y la capacidad de adaptación continua a los requerimientos cambiantes que caracterizan los proyectos de desarrollo de software contemporáneos.

ráneos. Esta adaptabilidad metodológica constituye un factor crítico para el éxito de proyectos en entornos tecnológicos dinámicos y de alta incertidumbre.

En el ámbito académico y específicamente en el desarrollo de software educativo, las metodologías ágiles adquieren una relevancia particular. Según [15], estas metodologías permiten superar las limitaciones de los enfoques “pesados”, tradicionalmente utilizados en la ingeniería del software educativo, los cuales resultaban poco adecuados para proyectos de pequeña escala y con recursos limitados. Las metodologías ágiles ofrecen procesos más manejables, facilitan la planificación flexible y permiten una mayor adaptación al cambio, enfatizan el valor del factor humano y la colaboración continua. En el contexto educativo, esto se traduce en la participación activa del docente dentro del equipo de desarrollo, lo cual facilita la validación temprana del producto mediante prototipos funcionales y mejora la alineación entre las necesidades y las funcionalidades implementadas.

2.1.5. Sistemas de información

Tal como nos explica [16] los sistemas de información son un conjunto de componentes interrelacionados los cuales recolectan, procesan, almacenan y distribuyen información. El propósito principal de estos sistemas es apoyar a la toma de decisiones y de control de una organización. En adición puede apoyar a los gerentes y trabajadores a analizar problemáticas visualizando temas complejos y crear nuevos productos.

Para poder transformar los datos en bruto en información de utilidad debemos pasar por un proceso con tres actividades básicas:

- **Entrada:** Recolecta los datos en bruto.
- **Procesamiento:** Convierte estos datos en bruto en un formato significativo.
- **Salida:** Transfiere la información procesada a las personas que la utilizan, el autor de [16] nos indica que también se requiere de la retroalimentación que viene a ser la salida que devuelven los miembros de la organización.

Para comprender correctamente la idea debemos tener en claro la definición de datos e información. Los datos son un flujo de elementos los cuales representan eventos que ocurren en la organización, estos elementos se encuentran aún sin ningún orden o interpretación. Por otro lado la información son los datos ordenados e interpretados los cuales resultan significativos y útiles para los seres humanos.

2.1.6. Framework Laravel

Laravel es un framework de desarrollo web basado en PHP ampliamente adoptado en la industria debido a su enfoque en la eficiencia, la modularidad y la seguridad. Según Subecz en [17], Laravel se posiciona como uno de los frameworks PHP más funcionales gracias a su arquitectura bien estructurada, su sintaxis clara y su capacidad para acelerar la construcción de aplicaciones web modernas.

Desde el punto de vista arquitectónico, Laravel implementa el patrón Modelo–Vista–Controlador (MVC), lo que permite una organización modular del software, una separación clara de responsabilidades y un mantenimiento más eficiente del código. De acuerdo a lo visto en [18], el uso del patrón MVC reduce significativamente la complejidad del desarrollo al favorecer la flexibilidad, la reutilización del código y la escalabilidad del sistema. Los autores también identifican que Laravel resulta preferible para proyectos de gran escala que requieren tiempos de entrega más rápidos y equipos de desarrollo con recursos limitados.

Laravel integra un conjunto robusto de herramientas que facilitan el trabajo de los desarrolladores, tales como el motor de plantillas Blade, el sistema de ruteo, Eloquent ORM para el mapeo objeto–relacional, el mecanismo de migraciones de base de datos y el middleware para gestionar la capa lógica de las solicitudes HTTP. En [19] destacan que estas características permiten una alta productividad y lo vuelven especialmente adecuado para aplicaciones complejas que requieren operaciones avanzadas a nivel de servidor.

En términos de seguridad, Laravel incorpora mecanismos nativos para mitigar amenazas comunes en aplicaciones web, incluyendo ataques de falsificación de solicitudes entre sitios (CSRF), scripting entre sitios (XSS) e inyección SQL [17]. Estas medidas cumplen con las prácticas recomendadas por los estándares modernos y refuerzan la fiabilidad del framework en sistemas de misión crítica.

Los estudios comparativos también han evaluado el rendimiento de Laravel. En [18] demostraron mediante pruebas de benchmarking con Apache Benchmark que Laravel fue capaz de manejar aproximadamente 3000 peticiones por segundo, registrando además el menor tiempo de respuesta entre los frameworks evaluados (4.46 ms). Asimismo, Laravel mostró el menor consumo de memoria (518 KB) frente a alternativas como Symfony. Aunque requiere un mayor número de archivos para su ejecución, su desempeño global fue superior.

Por su parte, [19] evidencio que antes de aplicar técnicas de optimización, Laravel presen-

tó tiempos de respuesta más elevados y menor throughput en comparación con CodeIgniter y Symfony; no obstante, mantuvo una estabilidad del 100 % sin errores durante todas las pruebas. Tras aplicar optimizaciones como cacheo de rutas, carga ansiosa (eager loading), cacheo de vistas y mejoras en la base de datos, Laravel mostró la mayor mejora entre todos los frameworks evaluados, reduciendo su tiempo de respuesta hasta en un 35 % bajo escenarios de alta carga. Esta capacidad de optimización convierte a Laravel en una opción altamente competitiva para sistemas de comercio electrónico y aplicaciones de alta demanda.

Finalmente, en [18] aplicaron el método QSOS para evaluar la madurez y calidad del framework, otorgando a Laravel puntuaciones máximas en estabilidad, comunidad activa, documentación, características técnicas y facilidad de uso. Dado su origen como un fork de Symfony, Laravel hereda estándares modernos y se complementa con servicios, soporte comunitario y herramientas de formación bien consolidadas.

En conjunto, estos estudios confirman que Laravel es un framework maduro, seguro y altamente productivo, especialmente adecuado para aplicaciones complejas, sistemas empresariales y plataformas web de gran escala que requieren un balance entre facilidad de desarrollo, rendimiento y capacidad de optimización.

2.1.7. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente reconocido por su robustez, fiabilidad y extensibilidad. A nivel conceptual, se define como un open source relational database management system (RDBMS) orientado a objetos, diseñado inicialmente como un proyecto académico en la Universidad de California, Berkeley [20]. Su evolución histórica y su modelo de licenciamiento han permitido que la comunidad y las organizaciones lo adopten como una alternativa madura y estable frente a soluciones propietarias.

- **Arquitectura y Naturaleza del Sistema:** PostgreSQL opera bajo el modelo cliente/servidor, en el que el proceso principal (postmaster) gestiona las solicitudes de conexión y crea procesos dedicados por cada cliente [21]. Este modelo asegura un manejo eficiente de múltiples sesiones concurrentes y una comunicación fiable entre las aplicaciones cliente y el servidor de base de datos.

2.1.8. Bootstrap

Según [22] Bootstrap es un marco (framework) de código abierto para el desarrollo web front-end, diseñado para la creación de sitios web receptivos (responsive). Fue lanzado por Twitter en 2011 para abordar la inconsistencia y los problemas de diseño de interfaz. Bootstrap es una librería que utiliza HTML, CSS y JavaScript, y se enfoca en el diseño "mobile-first" (móvil primero), asegurando que las páginas web se reajusten según el tamaño de la pantalla del dispositivo. Es popular entre los desarrolladores porque ofrece un conjunto gratuito de herramientas y plantillas prediseñadas, acelerando el proceso de desarrollo. Sus características clave incluyen componentes de interfaz de usuario preconstruidos y un potente sistema de cuadrícula receptiva de 12 columnas. Esta capacidad de adaptación y personalización lo convierte en el marco CSS front-end más descargado y popular.

2.1.9. HTTP Apache

En [23] nos indican que el Proyecto Servidor HTTP Apache (The Apache HTTP Server Project) es un esfuerzo colaborativo de desarrollo de software que ha creado un paquete de software de servidor HTTP robusto y rico en funciones. Es gestionado por el Grupo Apache, un conjunto de voluntarios distribuidos geográficamente que utilizan Internet para desarrollar y distribuir el servidor y su documentación relacionada. El origen del proyecto se remonta a febrero de 1995, cuando un pequeño grupo de webmasters se coordinó a través de Internet para unificar sus extensiones y correcciones de errores.

2.1.10. Arquitectura de Software (MVC - Modelo Vista Controlador)

Según [24] el Modelo-Vista-Controlador es uno de los patrones arquitectónicos más utilizados en el desarrollo de software. Es considerado un patrón arquitectónico porque se encarga de definir la estructura básica de un sistema, sirviendo como una plantilla base sobre la cual se construye una aplicación.

El patrón MVC propone dividir la aplicación en tres módulos claramente diferenciados: el modelo, la vista y el controlador. Cada uno de estos módulos tiene una funcionalidad definida:

- **El Modelo:** Es el encargado de la gestión de la información. Almacena representaciones abstractas de la información que el sistema va a manejar.

- **La Vista:** Se encarga de presentar la información a los usuarios finales. También se encarga de recoger la entrada del usuario (o su interacción con el sistema).
- **El Controlador:** El Controlador actúa como el intermediario y director del flujo de la aplicación.

2.1.11. Patrones de Diseño

Como nos indican en [25] un patrón como un modelo que explica cómo resolver un determinado problema. Es crucial entender que un patrón es un concepto o el razonamiento lógico para resolver un problema, no un conjunto de herramientas implementadas en el código de una aplicación. Estos patrones son estructuras bien definidas que permiten mantener una lógica de organización en el código de un sistema, evitando duplicaciones de código y facilitando su reutilización. Al utilizarlos, se puede crear software de calidad con mayor facilidad de mantenimiento y mejor comprensión del código al buscar modularidad en el sistema.

2.1.12. Estilo arquitectónico REST

El estilo arquitectónico REST (Representational State Transfer) ha sido ampliamente descrito como un modelo fundamental para el diseño de servicios web modernos. Según se explica en [26], REST fue caracterizado en la tesis de Roy Fielding (2000), donde se analiza la Web como caso de referencia y se resalta el papel central de los recursos como entidades identificables y manipulables mediante representaciones. A diferencia de los enfoques orientados a objetos o basados en RPC, REST propone un modelo sustentado en interfaces uniformes, operaciones HTTP con semántica simple y la transferencia explícita del estado.

El documento [26] destaca que REST es un estilo compuesto, derivado de la integración de múltiples estilos básicos como cliente-servidor, sistemas en capas, arquitectura sin estado, uso de caché, repositorios replicados y código a demanda. Esta composición permite lograr sistemas escalables, interoperables y de bajo acoplamiento.

En su clasificación, REST aparece dentro de los estilos peer-to-peer, así como dentro de las arquitecturas basadas en recursos, lo que refuerza su naturaleza híbrida. Además, [26] señala que REST influyó la evolución de los Web Services al desplazar la visión tradicional de “objetos distribuidos” hacia interacciones centradas en el intercambio de documentos y en restricciones que mantienen la coherencia del modelo, como la ausencia de estado y la

simplificación del uso de HTTP.

2.1.13. API REST

Una API REST (Application Programming Interface REST) es un tipo de API web que se fundamenta en un conjunto de principios que guían su creación, mantenimiento y escalabilidad. Están diseñadas para ser simples y fáciles de usar. Se caracterizan por utilizar métodos y códigos de estado HTTP estándar y están pensadas para ser escalables, flexibles, ligeras, rápidas y eficientes. Son independientes de la plataforma, lo que permite que cualquier cliente o dispositivo capaz de enviar y recibir solicitudes HTTP pueda utilizarlas [27].

2.1.14. Seguridad en Aplicaciones Web

Se considera la seguridad como el propósito principal de quienes administran recursos y datos, con el fin de protegerlos frente a posibles atacantes. Las organizaciones demandan que los profesionales de tecnologías de la información desarrollen aplicaciones que garanticen la integridad, confidencialidad y disponibilidad de la información. Dado que todo sistema, sitio o aplicación web es susceptible de sufrir ataques informáticos, resulta indispensable implementar mecanismos y técnicas de defensa que permitan prevenir intrusiones y asegurar la protección de los datos de los usuarios [28].

2.1.15. Sistemas de Gestión Documental (DMS)

Los sistemas de gestión documental según [29] son definidos como un tipo de sistema de información el cual garantiza una gestión documental digital eficiente, estos sistemas almacenan los documentos y los preservan. Los DMS nos dan la funcionalidad de archivar, almacenar y editar, crear, catalogar y proteger documentos para que se puedan recuperar de manera sencilla.

2.1.16. Sistemas de Gestión Documental en Instituciones educativas

Tal como se pudo observar en el apartado de antecedentes numerosas instituciones educativas implementaron sistemas de gestión documental [9], [5], [4], [3]. Si bien en algunos casos formaron parte de un sistema académico o se limitaron a un tipo específico de documento pudimos observar que en cada uno de los casos de estudio se encontraron con resultados favorables. Estos sistemas han sido adoptados tanto en instituciones de educación básica e instituciones de educación superior.

Nuestro proyecto busca adaptar el sistema de gestión documental con las definiciones claves especificadas en [30] las cuales son: competencias, capacidades, estándares de aprendizaje y desempeños. El objetivo es integrar estos conceptos en el sistema para poder reducir la carga administrativa de los docentes, reducir la generación de documentos físicos los cuales son propensos a perdidas y mejorar la trazabilidad de los documentos que sustentan dichas definiciones.

2.1.17. Marco Curricular Nacional de la Educación Básica del Perú

El Marco curricular Nacional de la Educación Básica del Perú es un instrumento de la política educativa en el Perú. El objetivo principal es establecer los aprendizajes que se espera logren los estudiantes, estos aprendizajes están definidos por criterios los cuales establecen rutas de aprendizaje las cuales respetamos la diversidad social, cultural, biológica y geográfica del Perú. La estructura definida en [30] se tienen cuatro definiciones curriculares claves: competencias, capacidades, estándares de aprendizaje y desempeños; estos conceptos son fundamentales para nuestro sistema web propuesto debido a que se modelará su arquitectura de información en torno a estos conceptos, el modelado de estos conceptos estarán en nuestra base de datos pudiendo así ser fácilmente recuperados y organizados así veremos alineado al flujo de información de los docentes pudiendo mejorar la eficacia y organización de los aprendizajes sin la necesidad de introducir nuevos conceptos a los docentes.

2.1.18. Planificación curricular y de áreas

La planificación curricular de los áreas en la Educación Básica se estructura en torno a cuatro definiciones claves según nos indica [30]: competencias, capacidades, estándares de

aprendizaje y desempeños. A continuación detallaremos cada uno de estos conceptos y su funcionalidad dentro del plan curricular.

- **Competencias:** Se define como la facultad de una persona para poder combinar un conjunto de capacidades con el fin de lograr un propósito definido en una situación en específico actuando de forma pertinente y con sentido ético. Para poder llegar a ser competente se necesita comprender la situación, evaluar las posibilidades de resolución, identificar conocimientos, identificar las habilidades a su disposición, analizar el plan de acción más pertinente, tomar la decisión y ejecutarla. Las competencias se construyen de forma constante, deliberada y conscientemente, siendo impulsadas por docentes e instituciones educativas a lo largo de la vida.
- **Desempeños:** Son descripciones específicas y observables de lo que los estudiantes hacen en relación con el desarrollo de las competencias, estas nos ayudan a medir el nivel esperado de la competencia. Dentro de la planificación y evaluación le ayuda a los docentes a reconocer la diversidad de los niveles de desempeño del aula brindando flexibilidad.
- **Capacidades:** Son los recursos que permiten actuar de manera competente, estas incluyen los conocimientos, habilidades y actitudes que los estudiantes usan para afrontar las diferentes situaciones. Según nos indica [30] el aprendizaje de conocimientos es un proceso vivo, no una repetición mecánica.
- **Habilidades:** Talentos, pericias o aptitudes para desarrollar tareas con éxito, que pueden ser sociales, cognitivas o motoras.
- **Actitudes:** Disposiciones o tendencias para actuar de acuerdo o en desacuerdo con una situación específica, que son formas habituales de pensar, sentir y comportarse según un sistema de valores.
- **Estándares de Aprendizaje:** Son descripciones del desarrollo de las competencias en niveles de creciente complejidad. Define el nivel que se espera que los estudiantes alcancen al finalizar cada ciclo. Estas descripciones articulan las capacidades de resolver situaciones de la vida real. Los estándares de aprendizaje sirven como referencia para la evaluación y para ayudar a identificar la cercanía o lejanía de los estudiantes con respecto al nivel esperado pudiendo así de esta forma servir como un referente al momento de la programación de las actividades y la elaboración de materiales educativos.

En resumen la planificación curricular se basa en el desarrollo de competencias a través de la movilización de capacidades. Los estándares de aprendizaje definen los niveles esperados de estas competencias, y los desempeños ofrecen descripciones detalladas de cómo se manifiestan estas competencias en la práctica, sirviendo de guía para docentes en la creación de experiencias de aprendizaje y en la evaluación formativa.

2.1.19. Documentos académicos de la institución educativa Padre Pérez De Guereñu

La institución educativa sigue con la planificación definida en [30] en este apartado se definen los documentos trabajados por la institución educativa Padre Pérez De Guereñu.

- **Diversificación Curricular:** La diversificación curricular es un proceso que busca adaptar el currículo a la diversidad de los estudiantes y a las necesidades y demandas de cada región. La Diversificación Curricular se basa en la premisa de que todos los estudiantes peruanos tienen derecho a aprendizajes comunes, pero también a aprendizajes diferenciados que respondan a sus propias realidades. Esta diversificación fue adoptada por la institución educativa como un documento físico el cual se realiza para cada grado y sección por área enseñada, siendo de autoría del docente asignado, en este documento tenemos definidos todas las competencias las cuales serán tocadas, cada una de estas competencias tiene especificadas las capacidades, desempeños y campos temáticos a tocar. También se definen las competencias transversales con sus capacidades y desempeños, las orientaciones metodológicas (métodos, técnicas o dinámicas grupales y técnicas gráficas esquemáticas) se definen en este documento junto con los lineamientos de evaluación, estos lineamientos de evaluación se encuentras divididos en Diagnóstica, Formativa y sumativa.
- **Planificación Anual:** La planificación anual al igual que la diversificación curricular es otro documento físico el cual se realiza para del área enseñada para cada grado y sección con la diferencia que no representa un concepto que podamos encontrar en [30] sino se centra en organizar el desarrollo de competencias comunicativas en los estudiantes. En este documento se definen las unidades de aprendizaje con sus respectivas situaciones significativas, cada una orientada a fortalecer las capacidades de los estudiantes. Asimismo, se establecen las competencias transversales. Finalmente, el documento especifica los criterios de evaluación diagnóstica, formativa y sumativa, junto con los instrumentos que permitirán evidenciar (bibliografía) los logros alcanzados.
- **Unidades Didácticas (Unidades de Aprendizaje):** Las unidades didácticas organizan los procesos educativos en períodos definidos y articulan la planificación anual con la práctica pedagógica. Según el [30], su finalidad es garantizar tiempo suficiente para el desarrollo progresivo de las competencias. Estas unidades se estructuran en torno a situaciones significativas, que constituyen experiencias reales o simuladas con sentido para los estudiantes. Dichas situaciones retan las competencias, despiertan el interés y promueven la movilización de capacidades cognitivas, procedimentales y actitudinales, consolidando así el enfoque por competencias.

- **Sesiones de Aprendizaje (Orientaciones para la Ejecución):** Las sesiones de aprendizaje son el nivel operativo donde se concretan las orientaciones pedagógicas para el desarrollo de competencias. En ellas, el docente implementa las situaciones significativas mediante estrategias activas como aprender haciendo, recuperar saberes previos, generar conflicto cognitivo y fomentar el trabajo cooperativo. Asimismo, las sesiones constituyen el espacio central para la evaluación formativa, que permite valorar el desempeño, identificar avances y brindar retroalimentación oportuna para la mejora continua del aprendizaje.

2.2. Objetivos del Proyecto

2.2.1. Objetivo General

Desarrollar un software de gestión documental para los docentes de la Institución Educativa Padre Pérez de Guereñu que centralice, organice y facilite el seguimiento de sus documentos, planificación de áreas y horarios.

2.2.2. Objetivos Específicos

1. Analizar los requerimientos funcionales y no funcionales de un sistema de gestión documental orientado a docentes.
2. Analizar el flujo de trabajo y los documentos utilizados por los docentes y personal encargado de la asignación de horarios y áreas a dictar.
3. Implementar funcionalidades para el registro, clasificación y búsqueda de documentos académicos y administrativos.
4. Evaluar la usabilidad y desempeño del prototipo con los docentes para validar eficiencia y satisfacción.

2.3. Justificación

La gestión administrativa y académica de las instituciones educativas peruanas son un problema crítico el cual buscamos abordar con la creación de este software. Los docentes

enfrentan una carga administrativa considerable la cual deriva de la gestión manual de documentos académicos como planificaciones y registros de evaluación. Según [10] el 84.4 % de los docentes peruanos reportan sentir estrés debido a la gestión administrativa de las instituciones educativas; la asignación de horarios y administración de carga laboral en los docentes son unas de las razones las cuales los docentes sienten dicho estrés, debido a que dificulta las labores pedagógicas disminuyendo el tiempo dedicado a las mismas.

La transformación digital es una prioridad para el sector educativo en el Perú como nos indica en [2], según los planes de modernización institucional. Este estudio resulta oportuno porque se alinea con las políticas de digitalización y mejora de la gestión escolar. Además, atiende la necesidad de los docentes de la Institución Educativa Padre Pérez de Guereñu, siendo clave en la mejora de resultados de aprendizaje. Aprovecharemos avances tecnológicos accesibles como sistemas de gestión de base de datos de código abierto y frameworks web (Laravel) que permiten un despliegue rápido y escalable.

En el ámbito de la ingeniería de software, se propone un caso de estudio completo que abarca las etapas de análisis, diseño, implementación y validación, aportando buenas prácticas y lecciones aprendidas que aportan al desarrollo de soluciones tecnológicas; además en el contexto educativo, esta solución nos brinda una mayor facilidad a la trazabilidad de documentos y ayuda a el uso de estándares en los formatos y la reducción de errores humanos, aliviando el estrés de los docentes mediante la automatización de cargas administrativas complejas, como la asignación de areas; para la comunidad académica, este trabajo genera un artefacto replicable en otras instituciones educativas y establece un marco de referencia que puede orientar futuros desarrollos de software educativo, promoviendo una gestión más eficiente y alineada con las necesidades de los colegios del Perú.

La implementación de nuestro sistema ayudará a que la Institución Educativa Padre Pérez de Guereñu adopte un flujo de trabajo integrado y reduzcan el uso de papel. Asimismo los docentes podrán visualizar y gestionar sus documentos junto con la planificación de areas y horarios desde una interfaz unificada. Con estas funcionalidades podremos recopilar métricas de uso y rendimiento. El desarrollo del sistema aplicará la metodología ágil Scrum [31] y técnicas de ingeniería de requisitos (entrevistas, casos de uso) [12] que demuestran cómo conducir un proyecto de desarrollo de software en un entorno educativo. Facilitando la validación temprana de funcionalidades con usuarios reales (docentes).

2.4. Viabilidad

El presente proyecto cuenta con una alta viabilidad, tanto económica, tecnológica y operativamente debido a una correcta planificación en trabajo conjunto con la Institución Educativa Padre Pérez de Guereñu.

La propuesta ha sido diseñada para tener el menor impacto financiero posible en la institución educativa. El primer punto tocado fue el costo del desarrollo del software el cual será asumido por ambos tesis en su totalidad como parte del proyecto de grado que esta siendo presentado actualmente, esto resultó en la eliminación total de costos asociados con contratación de personal o terceros para el desarrollo. Se implementará una solución de servidores locales los cuales nos brindará la institución educativa. Estos servidores anularán el gasto por servicios de alojamiento en la nube. Las tecnologías seleccionadas para el proyecto son el framework Laravel/PHP como gestor de base de datos PostgreSQL y JavaScript; todas las tecnologías seleccionadas son de código abierto y de uso libre lo cual libra a la institución educativa de cualquier costo extra relacionado con la adquisición de cualquier licencia para el funcionamiento del sistema. El único costo que será afrontado por la institución educativa será el pago del nombre de dominio web el cual es insignificante en comparación a los beneficios brindados por el software.

Todas las tecnologías seleccionadas cuentan con un amplio soporte brindado por la comunidad, una extensa documentación y han sido usados en múltiples proyectos de creación de aplicaciones web seguras y escalables.

Se hará un desarrollo a medida de la Institución Educativa Padre Pérez de Guereñu, lo cual asegura una adaptación perfecta al flujo de trabajo y las necesidades de los docentes y personal administrativo. El sistema está desarrollado con el fin de integrarse de forma natural al flujo de trabajo de los integrantes de la institución. La plataforma ayudará a digitalizar y centralizar los procesos que hasta el momento se han realizado de forma manual lo cual conlleva que estos procesos eran propensos a errores o estar desorganizados; con la implementación del software en el flujo de trabajo se reducirá la carga administrativa de los docentes y minimizará el riesgo de extraviar documentos.

La institución educativa tiene un control total sobre la infraestructura física y la seguridad de los datos debido a que la aplicación se encuentra alojada en el servidor local. Ambos puntos son puntos críticos de seguridad al manejar información sensible de docentes y planificación académica. Se diseñará un plan de recuperación ante desastres el cual consiste en la realización

Tabla 1: Desglose de Costos y Ahorros

Acción	Tipo	Descripción	Costo (S/)
Dominio	Costo	Registro anual de dominio web (.edu.pe) para la institución	80.00
Personal	Ahorro	Desarrollo por tesis (equivalente a 328 horas de trabajo profesional)	12,464.00
Hosting	Ahorro	Uso de servidores locales vs. cloud (equivalente a 2 instancias AWS)	3,600.00
Licencias	Ahorro	Algoritmos propios vs. soluciones comerciales (ej: Kronos Workforce)	8,000.00
Certificado SSL	Ahorro	Certificado SSL para conexión segura	300.00
Capacitación	Ahorro	Videos tutoriales vs. capacitación presencial externa	500.00
Backup	Costo	Disco adicional 4TB para respaldo local	600.00
Total		Balance final	38,780.00

de copias de seguridad periódicas en un servidor secundario. Con esta acción nos aseguramos que la integridad quede intacta y que se pueda tener una rápida restauración ante cualquier incidente de hardware o software.

2.5. Limitaciones

El funcionamiento correcto del sistema está directamente ligado a la disponibilidad del servidor local el cual podría sufrir interrupciones en el suministro de energía eléctrica o fallas de hardware en el servidor, esto provocaría que software se encuentre fuera de servicio temporalmente hasta que se pueda solucionar el inconveniente con el servidor local y se pueda brindar nuevamente el servicio al personal de la institución educativa.

Se debe tener en consideración que la solución será diseñada y adaptada al flujo de trabajo de la Institución Educativa Padre Pérez de Guereña, junto con los documentos que la misma trabaja; por lo tanto la implementación de ese software podría ser tomado como una guía para próximas implementaciones en distintas instituciones educativas teniendo que realizar una mínima personalización del flujo o de los documentos diseñados en el sistema. Toda modificación de los documentos y del flujo del sistema se deben realizar directamente en el

código fuente de la aplicación.

Se garantiza el soporte y mantenimiento brindado a la institución sobre el software mientras se esté en desarrollo la tesis. Se debe definir un responsable en la institución para asegurar la operatividad y mantenimiento.

Capítulo III - METODOLOGÍA DE DESARROLLO

3.1. Plan de Gestión del Proyecto

3.1.1. Alcance

Buscando garantizar el éxito del proyecto junto con la entrega de una solución funcional la cual se adapte a las necesidades de la institución educativa Padre Pérez de Guereñu es fundamental precisar el alcance. En el presente apartado se definirán los procesos, funcionalidades y áreas que serán abordadas; también se definirán los límites de nuestro proyecto.

3.1.1.1. Alcance del proyecto El proyecto abarca el ciclo de vida completo del desarrollo de un sistema web, desde el diseño y toma de requerimientos hasta la puesta en funcionamiento, el sistema web propuesto incluirá las siguiente funcionalidades y componentes:

- Sistema Central y Arquitectura:
 - Se desarrollará una aplicación web centralizada, la cual será accesible a través de navegadores web modernos en computadoras de escritorio y gracias al nuestro diseño responsivo también se podrá acceder desde dispositivos móviles.
 - La aplicación web se encontrará alojada en los servidores locales los cuales serán proporcionados por la institución Educativa Padre Pérez de Guereñu.
- Módulo Académico (Gestión Documental):
 - Registro y Almacenamiento: Capacidad del sistema para que los docentes puedan cargar, almacenar y gestionar de forma ordenada y seguro sus documentos pedagógicos, como planificaciones anuales, sesiones de aprendizajes y materiales de clase.
 - Versionado de Documentos: Funcionalidad para mantener un historial de las versiones que van cambiando de año en año, permitiendo a los docentes acceder a sus documentos antiguos.

- Búsqueda y Clasificación: Se implementará un motor de búsqueda que permita filtrar y localizar documentos por docente, curso o fecha.
- Módulo Administrativo (Planificación y Gestión):
 - Gestión de Areas y Docentes: Interfaz para que el personal administrativo pueda administrar los areas, grados, secciones y perfiles de los docentes de la institución.
 - Planificación de Horarios: Desarrollo de una interfaz visual de calendario para la creación, consulta y modificación de los horarios de clase.
 - Asignación de Carga Académica Automática: Herramienta para asignar automáticamente los docentes en los diferentes areas y horarios de acuerdo a su disponibilidad y carga de trabajo, esta funcionalidad será implementada por un algoritmo de autoría propia la cual sugerirá una distribución de horarios optimizada considerando las restricciones de disponibilidad del docentes y las áreas de conocimiento.
- Gestión de Usuarios y Seguridad
 - Autenticación y Roles: El sistema contará con una gestión flexible debido a que se basa en roles y permisos de escritura, lectura y edición y eliminación junto con permisos para la visualización de los datos completos, estos diferentes permisos podrán ser asignados a los diferentes roles necesarios que surjan con el tiempo en la institución educativa; actualmente se detectó que solo se hará uso del rol de Coordinador/Administrador y Docente.
 - Plan de Recuperación: Se realizarán copias de seguridad mensualmente contando con solamente con el último backup como histórico debido a las limitaciones del almacenamiento del servidor. Esta configuración busca garantizar la integridad y recuperación de la información.
- Procesos y Organismos Involucrados
 - Docentes de la Institución Educativa Padre Pérez de Guereñu: Serán los usuarios finales principales, con la ayuda de los docentes podremos realizar el levantamiento de requisitos y validación del sistema.
 - Personal Administrativo/Coordinación: Participarán en la definición de los flujos de trabajo administrativos y en la validación del módulo correspondiente.

3.1.1.2. Exclusiones del Proyecto Con el objetivo de adoptar el enfoque más adecuado a las necesidades de la institución educativa y asegurar la viabilidad de nuestro proyecto de tesis se definirán las exclusiones del proyecto en el siguiente apartado.

- Portal para Estudiantes y Padres de Familia: El sistema esta diseñado exclusivamente para el uso del personal interno de la institución educativa. No se desarrollará ningún tipo de

interfaz ni funcionalidad para los alumnos o sus apoderados.

- Módulos Financieros o Contables: No se incluirá ninguna funcionalidad relacionada con algún proceso económico, tales como podrían ser gestión de pensiones, matrículas, pagos o facturación.
- Integración con Pasarelas de Pago: El sistema no procesará transacciones monetarias en línea.
- Módulo de Calificaciones Detallado: La gestión de documentos como registros de evaluación no incluye la implementación de un módulo de notas para el cálculo de promedios, libretas de notas o estadísticas de rendimiento.
- Aplicación Móvil Nativa: El sistema será accesible mediante navegadores web modernos, incluidos los de dispositivos móviles, gracias a su diseño con interfaces responsivas. No está contemplado el desarrollo de una aplicación nativa para Android ni iOS.
- Adquisición de Infraestructura de Hardware: El proyecto se desarrollará asumiendo que la institución educativa proveerá el servidor local necesario para el despliegue. La compra de equipos está fuera del alcance.
- Soporte y Mantenimiento Post-Tesis: El soporte técnico y el mantenimiento del sistema por parte de los tesis se garantizan durante el periodo de desarrollo del proyecto. El mantenimiento y la sostenibilidad a largo plazo será responsabilidad de la institución.

3.1.2. Metodología de desarrollo

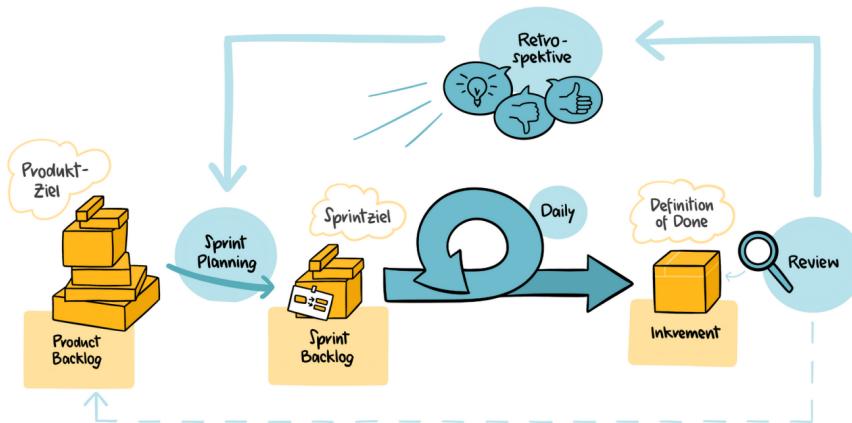


Figura 1: Metodología Scrum según [32]

Fuente: Imagen obtenida de Schwaber y Sutherland [32, pg. 20]

Para el siguiente estudio utilizaremos Scrum debido que tal como se indica en [31] Scrum no es una metodología rígida que dicte exactamente los pasos que se deben seguir sino un

framework. Esta flexibilidad es ideal para poder adaptarse a cada situación específica. Para el siguiente estudio utilizaremos Scrum por su flexibilidad y gran ayuda para poder medir y organizar y tener

3.1.2.1. Componentes y prácticas clave de Scrum

- Sprints: Un sprint es una iteración de duración de tiempo fija y corta, estas iteraciones deben ser menores a seis semanas según nos indica [31]. El objetivo de estos sprints es producir un incremento de software funcional.
- Product Owner: Es el “puente” el cual se encarga de transmitir los objetivos del equipo del producto y el equipo de desarrollo, es el que se encarga de representar al usuario o cliente, define la importancia y el alcance de las historias. Participa activamente en la planificación del alcance y la importancia.
- Scrum Master: No es solamente el intermediario entre el equipo de desarrollo y el Product Owner sino también es el facilitador que asegura que Scrum se esté siguiendo correctamente. Facilita las reuniones y se asegura que se estén cumpliendo con los tiempos establecidos manteniendo el Sprint Backlog actualizado.
- Scrum Master: No es solamente el intermediario entre el equipo de desarrollo y el Product Owner sino también es el facilitador que asegura que Scrum se esté siguiendo correctamente. Facilita las reuniones y se asegura que se estén cumpliendo con los tiempos establecidos manteniendo el Sprint Backlog actualizado.
- Development Team: Es el equipo responsable de desarrollar el producto, estima el esfuerzo de las historias y decide cuáles historias deben ser incluidas en cada Sprint basándose en la capacidad y las prioridades del Product Owner.
- Sprint Goal: Es el objetivo principal del Sprint, este objetivo debe tener un significado tanto para el equipo de desarrollo como para el product Owner para que así se puedan mantener enfocados y sin ningún tipo de confusión.

Dentro de cada uno de los Sprint tenemos unos eventos claves para el desarrollo del mismo

- Sprint Backlog: Es la lista de historias o también llamados requisitos que se realizarán en dicho Sprint, el encargado de crear estas Sprint Backlog son el equipo de desarrollo.
- Sprint Planning: Es una reunión de alta importancia debido a que como su nombre lo indica es la planificación de las actividades que se realizarán en el Sprint. El product owner y el equipo de desarrollo son los que deben asistir a estas reuniones para poder realizar un ajuste continuo de alcance, importancia y estimación.
- Sprint Planning: Es una reunión de alta importancia debido a que como su nombre lo indica

es la planificación de las actividades que se realizarán en el Sprint. El product owner y el equipo de desarrollo son los que deben asistir a estas reuniones para poder realizar un ajuste continuo de alcance, importancia y estimación.

- Scrum Diario: Es una reunión diaria de aproximadamente 15 minutos la cual se debe realizar en el mismo lugar y el mismo lugar, los objetivos principales son mantener al equipo sincronizado y enfocado con los objetivos del Sprint.
- Demo del Sprint: En esta reunión se debe presentar los objetivos del Sprint, estos objetivos deben ir de la mano con el código funcional, el objetivo principal de esta reunión es poder recibir el feedback de los interesados y poder mostrar el avance real del equipo.
- Retrospectiva del Sprint: Según [31] es el segundo evento más importante de Scrum debido a que nos brinda la mejor oportunidad de una mejora continua. El equipo reflexiona sobre aquello que salió bien, lo que pudo haber sido mejor y las cosas que deberían realizarse diferente en el próximo Sprint.

3.1.3. Planificación e historias de usuario

El proceso del trabajo se definió en el product backlog el cual se desglosó en 3 sprints. A las historias se les asignó un identificador único con una nomenclatura que busca mantener el orden y trazabilidad del proyecto. Este cronograma se realizó mediante un diagrama de Gantt diseñado con la herramienta GanttProject, el cual se encuentra en el Anexo 1.

Tabla 2: Historias de Usuario y Componentes del Sistema (Parte 1)

ID	Epic/Componente	Nombre	Imp.	Estim.	Prueba	Notas	Dependencias
H001	Gestión Usuarios	Registro/inicio de sesión	200	3	CA001	Autenticación básica, sesiones seguras.	Ninguna
H002	Gestión Usuarios	Recuperar contraseña	120	2	CA002	Email en staging.	H001
H003	Gestión Usuarios	Gestión roles y permisos	180	5	CA003	Roles iniciales: Coordinador, Docente.	H001
H004	Gestión Documental	Subir documento pedagógico	190	3	CA004	Soporta PDF/DOCX/ZIP; tamaño límite.	H001, H006
H005	Gestión Documental	Estructura por año y carpeta	160	3	CA005	Estandarizar nombres.	H004
H006	Gestión Usuarios	Perfil de docente editable	140	2	CA006	Datos usados por asignación.	H001
H007	Versionado Documentos	Control de versiones de documento	170	5	CA007	Mostrar autor y fecha por versión.	H004, H005
H008	Gestión Documental	Restaurar versión anterior	130	3	CA008	Soft-restore.	H007
H009	Búsqueda y Clasificación	Búsqueda por docente/curso/fecha	180	5	CA009	Soportar búsqueda parcial.	H004, H005
H010	Búsqueda y Clasificación	Etiquetado / categoría de documentos	120	3	CA010	Etiquetas administrables.	H004
H011	Gestión Documental	Descarga segura de documento	150	2	CA011	Verificar permisos antes de descarga.	H001, H003, H004
H012	Gestión Documental	Eliminación (soft-delete) de documento	110	2	CA012	Política de retención configurable.	H004
H013	Gestión Académica	Crear curso/grado/sección	190	4	CA013	Campos: nombre, código, jornada.	H001

Tabla 3: Historias de Usuario y Componentes del Sistema (Parte 2)

ID	Epic/Componente	Nombre	Imp.	Estim.	Prueba	Notas	Dependencias
H014	Gestión Académica	Crear/editar perfil docente (admin)	160	3	CA014	Incluye áreas y carga máxima.	H001
H015	Planificación Horarios	Interfaz visual calendario (ver)	200	5	CA015	Diseño responsive.	H013, H014
H016	Planificación Horarios	Crear/editar horario manualmente	170	5	CA016	Undo/redo simple.	H015, H013
H017	Planificación Horarios	Detección de conflictos en horario	180	3	CA017	Validaciones configurables.	H016, H014
H018	Asignación Automática	Asignación automática de docentes (algoritmo)	200	8	CA018	Mostrar score y razones.	H013, H014, H015
H019	Asignación Automática	Aceptar/rechazar sugerencias automáticas	140	2	CA019	Soporte override manual.	H018
H020	Export / Reportes	Exportar horario a PDF/CSV	130	2	CA020	Plantilla institucional.	H015
H021	Infraestructura y Seguridad	Backup mensual automático	160	3	CA021	Solo último backup histórico.	H001
H022	Gestión Documental	Control de accesos por documento	190	4	CA022	ACL por usuario/rol.	H003, H011

Tabla 4: Criterios de Aceptación (Given-When-Then)

ID	Given (Dado)	When (Cuando)	Then (Entonces)
CA001	Soy docente o coordinador no autenticado.	Ingreso usuario y contraseña válidos y hago clic en login.	Accedo al dashboard y se aplica mi rol.
CA002	He olvidado mi contraseña.	Solicito reset de contraseña por correo.	Recibo el enlace y puedo cambiar la contraseña.
CA003	Soy Coordinador.	Creo o edito un rol y le asigno permisos.	Los usuarios con ese rol ven y pueden ejecutar según permisos.
CA004	Estoy autenticado como Docente.	Subo un archivo y completo los metadatos.	El documento aparece en mi repositorio con confirmación.
CA005	Estoy subiendo un documento e indico año y tipo.	Guardo el documento.	Se almacena en la ruta /Año/Docente/Tipo.

Continúa en la siguiente página

Tabla 4 – *Continúa de la página anterior*

ID	Given (Dado)	When (Cuando)	Then (Entonces)
CA006	Soy Docente.	Edito mis datos.	Los cambios se guardan y se muestran en mi perfil.
CA007	Existe un documento ya guardado.	Subo una nueva versión del documento.	Se crea un historial de versiones y puedo ver versiones previas.
CA008	Estoy en el historial de versiones de un documento.	Selecciono una versión y elijo restaurar.	La versión seleccionada se restaura como la versión activa (se crea registro).
CA009	Estoy en el módulo documental.	Aplico filtro por curso o fecha.	Los resultados coinciden con los filtros y se muestran ordenados.
CA010	Estoy creando o editando un documento.	Agrego una etiqueta/categoría al documento.	El documento aparece bajo esa etiqueta en los filtros.
CA011	Tengo permiso para acceder al documento.	Hago clic en descargar.	El archivo se descarga y la acción queda registrada en el log.
CA012	Soy Docente o Coordinador.	Elimino un documento desde la interfaz.	El documento se marca como borrado (soft-delete) y puede restaurarse desde la papelera.
CA013	Soy Coordinador.	Creo curso, grado y sección en el catálogo.	Las entidades creadas aparecen en el catálogo y pueden seleccionarse para asignaciones.
CA014	Soy Coordinador.	Creo o edito el perfil de un docente.	El perfil queda disponible para asignaciones y aparece en listados.
CA015	Soy Coordinador o Docente autenticado.	Abro la vista del calendario/horario.	Se muestra el horario por semana/día por secciones y docentes.
CA016	Soy Coordinador.	Arrastro o edito una franja en el calendario.	El horario se guarda y se validan conflictos básicos.

Continúa en la siguiente página

Tabla 4 – *Continúa de la página anterior*

ID	Given (Dado)	When (Cuando)	Then (Entonces)
CA017	Estoy creando o modificando una franja horaria.	Existe solapamiento o exceso de carga para un docente.	El sistema muestra una alerta y bloquea la acción salvo confirmación explícita.
CA018	Existe el catálogo de áreas y la disponibilidad de docentes.	Ejecuto el algoritmo de asignación automática.	El sistema propone una asignación optimizada y muestra la lista de sugerencias.
CA019	Hay sugerencias generadas por el algoritmo.	El administrador acepta o ajusta la sugerencia.	Los cambios se aplican y queda registrada la auditoría.
CA020	Estoy visualizando un horario.	Hago clic en exportar (PDF o CSV).	Se genera un archivo descargable con el layout imprimible.
CA021	El servidor local está operativo.	Llega la fecha programada de backup.	Se genera el backup y se reporta el estado (ok/fallo).
CA022	Existe un documento con permisos específicos.	Un usuario no autorizado intenta acceder al documento.	El acceso es denegado y el intento queda registrado en auditoría.

3.1.4. Recursos del Proyecto

Los recursos fueron divididos en cuatro principales categorías, en la categoría de “Recursos Humanos” se encuentra nuestra persona y el personal que nos ayudará a la recopilación requisitos y validación de los mismos, “Recursos Tecnológicos” se centra únicamente el hardware y software que nos ayudará al desarrollo y prueba de la aplicación web, “Recursos de Infraestructura” para que el proyecto pueda brindar el servicio a los usuarios finales y por último “Recursos de Comunicación” para las capacitaciones, documentación y diagramas desarrollados.

Tabla 5: Recursos del Proyecto (Parte 1)

Categoría	Recurso	Descripción	Cant.	Costo (\$/)	Responsable	Observaciones
Recursos Humanos	Tesista 1	Guillermo Jose Aleman Zambrano - Desarrollador Backend	1	0.00	Tesista	330 horas aprox.
	Tesista 2	Daniel Gustavo Mendi-guri Chavez - Desarrollador Frontend	1	0.00	Tesista	330 horas aprox.
	Docentes IE	Docentes para levanta-miento de requisitos y validación	13	0.00	IE Padre Pérez	Participación vo-luntaria
	Personal Administrativo	Coordinadores para de-finición de flujos	1-2	0.00	IE Padre Pérez	Participación en horario laboral
Recursos Tecnológicos	Laptops de Desarrollo	Equipos personales de los tesis-tas	2	0.00	Tesistas	Core i7, 16GB RAM
	Framework Laravel/PHP	Plataforma de desarro- llo web	1	0.00	Tesistas	Código abierto
	PostgreSQL	Sistema gestor de base de datos	1	0.00	Tesistas	Código abierto
	JavaScript/HTM-L/CSS	Tecnologías frontend	1	0.00	Tesistas	Estándares web
	Git/GitHub	Control de versiones	1	0.00	Tesistas	Repositorio pú-blico
	Visual Studio Co-de	Entorno de desarrollo	2	0.00	Tesistas	Editor gratuito

Tabla 6: Recursos del Proyecto (Parte 2)

Categoría	Recurso	Descripción	Cant.	Costo (\$)	Responsable	Observaciones
Recursos Tecnológicos (cont.)	Ambientes de Prueba	Entorno de staging	1	0.00	Tesistas	En equipos locales
	Herramientas de Testing	Para pruebas automatizadas	1	0.00	Tesistas	PHPUnit, Jest
	Dispositivos Móviles	Para pruebas de responsividad	1-2	0.00	Tesistas	Equipos personales
Recursos de Infraestructura	Servidor Local	Servidor proporcionado por la IE	1	0.00	IE Padre Pérez	i5 Gen 12, 32GB RAM, 1TB
	Dominio Web	Registro de dominio (.edu.pe opcional)	1	80.00	IE Padre Pérez	Renovación anual
	Disco de Backup	Almacenamiento adicional 4TB	1	600.00	IE Padre Pérez	Para respaldos
	Conexión a Internet	Acceso a internet estable	1	0.00	IE Padre Pérez	Recurso existente
Recursos de Comunicación	Plataforma de Videollamadas	Para reuniones remotas con docentes	1	0.00	Tesistas	Google Meet
	Herramientas de Documentación	Para documentación del proyecto	2	0.00	Tesistas	Google Docs
	Software de Diagramación	Para modelado UML y diagramas	1	0.00	Tesistas	Lucidchart, Draw.io, Umlertino

3.2. Análisis y Diseño del Sistema

3.2.1. Requerimientos del Sistema

Tabla 7: Requerimientos Funcionales (Parte 1)

ID	Tipo	Descripción	Pri- oridad	Criterio de Aceptación
RF-01	Gestión Usuarios	El sistema debe permitir el registro e inicio de sesión de usuarios con autenticación segura.	Alta	Un usuario iniciar sesión con credenciales válidas y mantener sesión activa y el administrador puede registrar nuevos usuarios.
RF-02	Gestión Usuarios	El sistema debe permitir la recuperación de contraseña mediante correo electrónico.	Me- dia	Un usuario puede solicitar recuperación y recibe enlace válido en su correo.
RF-03	Gestión Usuarios	El sistema debe gestionar roles y permisos (Coordinador, Docente, etc.).	Alta	El usuario solo puede acceder a funciones según su rol asignado.
RF-04	Gestión Documental	El sistema debe permitir subir documentos pedagógicos (PDF, DOCX, ZIP) con límite de tamaño.	Alta	Un docente sube un archivo válido y el sistema lo almacena correctamente.
RF-05	Gestión Documental	El sistema debe organizar documentos por año y carpeta con nombres estandarizados.	Me- dia	Los documentos se muestran clasificados y se ubican en la estructura definida.
RF-06	Gestión Usuarios	El sistema debe permitir a cada docente editar su perfil (datos no críticos).	Me- dia	Un docente puede actualizar sus datos y estos se reflejan en las asignaciones.
RF-07	Versionado Documentos	El sistema debe permitir el control de versiones mostrando autor y fecha.	Alta	Un documento actualizado refleja la versión más reciente y conserva historial.
RF-08	Versionado Documentos	El sistema debe permitir restaurar versiones anteriores de un documento.	Me- dia	El usuario selecciona una versión previa y el sistema la habilita nuevamente.
RF-09	Búsqueda	El sistema debe permitir búsquedas por docente, curso o fecha.	Alta	La búsqueda devuelve resultados parciales y exactos.
RF-10	Búsqueda	El sistema debe permitir etiquetar documentos con categorías personalizadas.	Me- dia	El usuario puede añadir, editar o eliminar etiquetas y clasificarlas.
RF-11	Gestión Documental	El sistema debe permitir la descarga segura de documentos verificando permisos.	Alta	Solo usuarios autorizados pueden descargar archivos.
RF-12	Gestión Documental	El sistema debe permitir la eliminación lógica (soft-delete) de documentos.	Me- dia	El documento eliminado no se muestra pero puede recuperarse dentro del periodo definido.
RF-13	Gestión Académica	El sistema debe permitir crear, actualizar y eliminar áreas, grados y secciones con sus datos básicos, asignándoles un código único.	Alta	El administrador puede crear, actualizar o eliminar áreas, grados y secciones, y los cambios se reflejan correctamente en el sistema.

Tabla 8: Requerimientos Funcionales (Parte 2)

ID	Tipo	Descripción	Pri- oridad	Criterio de Aceptación
RF-14	Gestión Académica	El administrador debe poder crear/editar perfiles de docentes con áreas y carga horaria.	Alta	El perfil se registra correctamente y es visible en asignaciones.
RF-15	Planificación Horarios	El sistema debe mostrar un calendario visual responsive para ver horarios.	Alta	El calendario se ajusta a dispositivos móviles y escritorio.
RF-16	Planificación Horarios	El sistema debe mostrar el calendario de acuerdo al rol del usuario (docente o administrador).	Alta	Un docente solo visualiza su propio calendario y el administrador puede visualizar los calendarios de todos los docentes.
RF-17	Planificación Horarios	El sistema debe permitir crear y editar horarios manualmente con opciones de undo/redo.	Alta	Los cambios realizados en un horario pueden deshacerse o rehacerse.
RF-18	Planificación Horarios	El sistema debe detectar conflictos de horarios y notificar al usuario.	Alta	Al guardar un horario con choque, el sistema advierte al usuario.
RF-19	Asignación Automática	El sistema debe asignar docentes a áreas de manera automática usando algoritmo.	Alta	El sistema muestra las asignaciones propuestas junto a un puntaje y razones.
RF-20	Asignación Automática	El usuario debe poder aceptar o rechazar las asignaciones automáticas y modificarlas manualmente.	Me- dia	El sistema registra la decisión final del coordinador.
RF-21	Exportación	El sistema debe exportar horarios a PDF y CSV con plantilla institucional.	Me- dia	El archivo generado cumple con formato institucional y se descarga correctamente.
RF-22	Seguridad	El sistema debe generar un backup mensual automático conservando solo el último.	Me- dia	El backup se ejecuta en la fecha configurada y es recuperable.
RF-23	Seguridad	El sistema debe permitir control de accesos a documentos mediante listas ACL por usuario/rol.	Alta	Los permisos de acceso se cumplen en cada intento de lectura o descarga.

3.2.2. Casos de Uso

■ RF-01: Gestión de Usuarios - Registro e Inicio de Sesión

Este caso de uso corresponde al requisito funcional **RF-01** (descrito en la Tabla 7). El sistema debe permitir la gestión de usuarios a través de dos funciones principales: el registro de nuevos usuarios realizado por un administrador y el inicio de sesión realizado por los usuarios con autenticación segura.

Prioridad: Alta.

Criterio de aceptación: Un usuario puede iniciar sesión con credenciales válidas y mantener la sesión activa, y un administrador puede registrar nuevos usuarios de forma correcta.

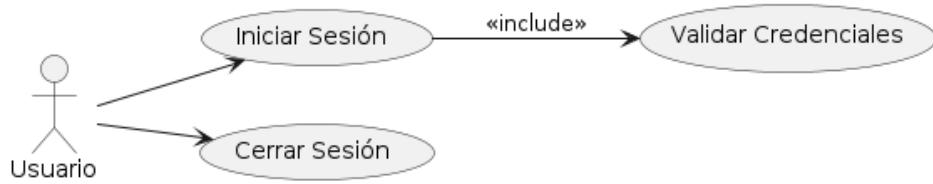


Figura 2: Caso de uso RF-01: Inicio de sesión

La Figura 2 ilustra el caso de uso de inicio de sesión. Asimismo, la Figura 3 muestra el caso de uso complementario de registro de usuario.

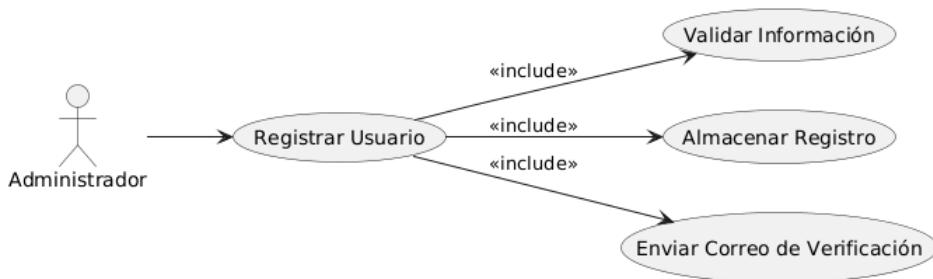


Figura 3: Caso de uso RF-01: Registrar usuario

■ RF-02: Gestión de Usuarios - Recuperación de Contraseña

Este caso de uso corresponde al requisito funcional RF-02 (descrito en la Tabla 7). El sistema debe permitir que un usuario recupere su contraseña ingresando su correo electrónico en la opción de recuperación. El sistema validará la existencia del correo asociado a una cuenta registrada y, en caso de ser válido, enviará un enlace seguro a la bandeja de entrada del usuario para restablecer la contraseña.

Prioridad: Media.

Criterio de aceptación: Un usuario puede solicitar la recuperación de su contraseña desde la plataforma, ingresando su correo electrónico. Si el correo es válido, el sistema envía un enlace seguro para restablecer la contraseña. Si el correo no está registrado, el sistema debe notificar que no existe una cuenta asociada.

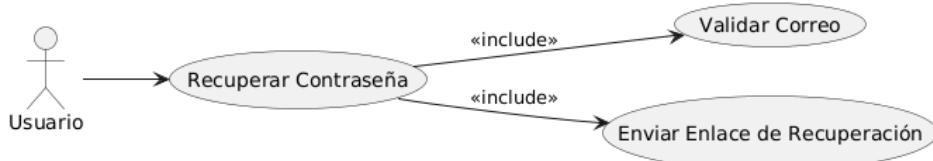


Figura 4: Caso de uso RF-02: Recuperación de contraseña

■ RF-03: Gestión de Usuarios - Roles y Permisos

Este caso de uso corresponde al requisito funcional **RF-03** (descrito en la Tabla 7). El sistema debe gestionar los roles y permisos de los usuarios (Coordinador, Docente, Administrador, etc.), asegurando que cada usuario solo pueda acceder a las funcionalidades que correspondan a su rol asignado.

Prioridad: Alta.

Criterio de aceptación: El usuario solo puede acceder a las funciones del sistema que estén habilitadas para su rol asignado, impidiendo el acceso a funciones no autorizadas.

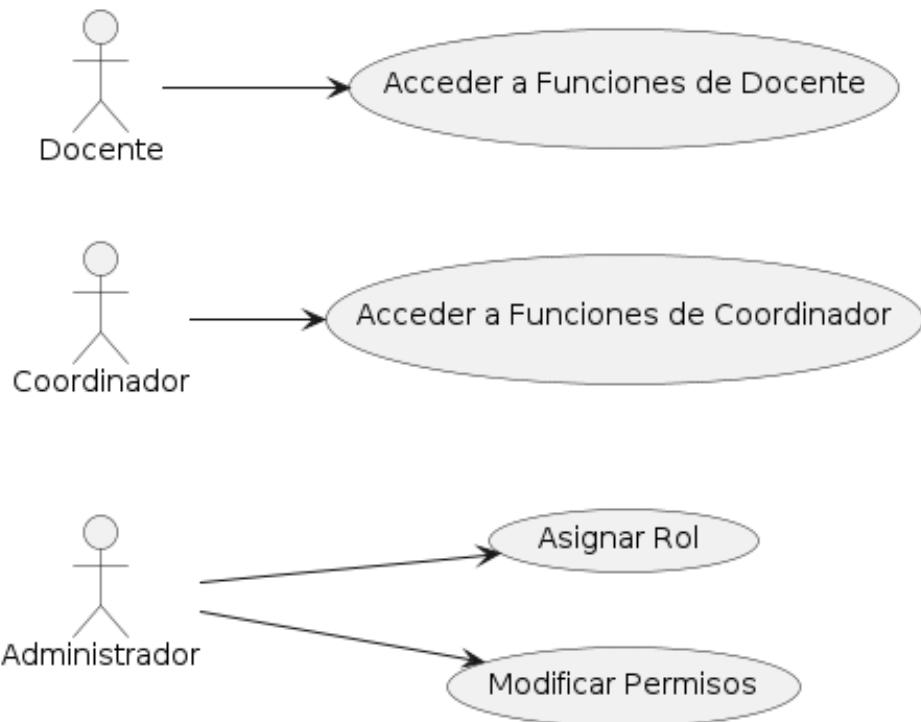


Figura 5: Caso de uso RF-03: Roles y permisos

■ **RF-04: Gestión Documental - Subida de Documentos**

Este caso de uso corresponde al requisito funcional **RF-04** (descrito en la Tabla 7). El sistema debe permitir que los docentes suban documentos pedagógicos en formatos admitidos (PDF, DOCX, ZIP), garantizando un control de tamaño máximo para su almacenamiento seguro en la plataforma.

Prioridad: Alta.

Criterio de aceptación: Un docente puede subir un archivo válido en formato permitido y con tamaño dentro del límite establecido, y el sistema lo almacena correctamente para su posterior consulta.

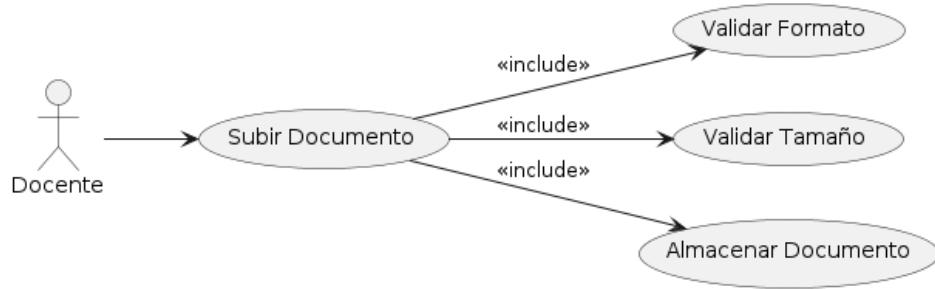


Figura 6: Caso de uso RF-04: Subida de documentos

■ RF-05: Gestión Documental - Organización de Documentos

Este caso de uso corresponde al requisito funcional **RF-05** (descrito en la Tabla 7). El sistema debe organizar los documentos pedagógicos de manera estructurada, clasificándolos por año y carpeta, y aplicando un esquema de nombres estandarizados para facilitar su búsqueda y consulta.

Prioridad: Media.

Criterio de aceptación: Los documentos se muestran clasificados correctamente por año y ubicados en la carpeta correspondiente con nombres estandarizados, siguiendo la estructura definida en el sistema.

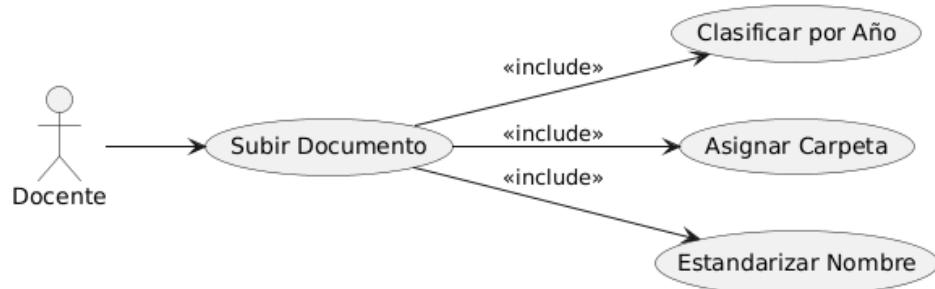


Figura 7: Caso de uso RF-05: Organización de documentos

■ RF-06: Gestión de Usuarios - Edición de Perfil

Este caso de uso corresponde al requisito funcional **RF-06** (descrito en la Tabla 7). El sistema debe permitir que cada docente edite su perfil, limitándose a la actualización de datos no críticos (por ejemplo, número de teléfono, correo alternativo o dirección). De esta manera, se asegura que la información relevante se mantenga actualizada sin comprometer aspectos de seguridad o gestión administrativa.

Prioridad: Media.

Criterio de aceptación: Un docente puede actualizar sus datos de perfil no críticos y los cambios se reflejan correctamente en el sistema, incluyendo sus asignaciones relacionadas.

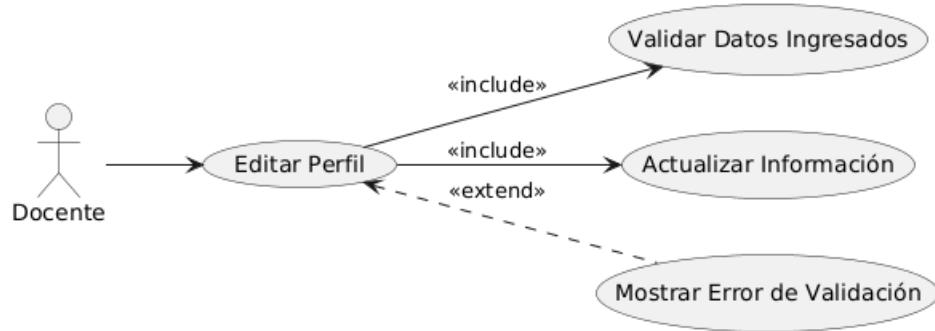


Figura 8: Caso de uso RF-06: Edición de perfil de docente

■ RF-07: Versionado de Documentos - Control de Versiones

Este caso de uso corresponde al requisito funcional **RF-07** (descrito en la Tabla 7). El sistema debe permitir el control de versiones de los documentos pedagógicos, mostrando para cada versión el autor que realizó la modificación y la fecha en que se efectuó. Además, debe garantizar que siempre se visualice la versión más reciente sin perder el historial de cambios.

Prioridad: Alta.

Criterio de aceptación: Un documento actualizado refleja la versión más reciente en el sistema y conserva un historial con autor y fecha de cada modificación realizada.

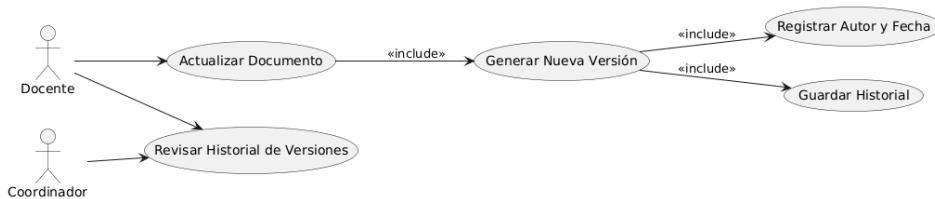


Figura 9: Caso de uso RF-07: Control de versiones de documentos

■ RF-08: Versionado de Documentos - Restauración de Versiones

Este caso de uso corresponde al requisito funcional **RF-08** (descrito en la Tabla 7). El sistema debe permitir a los usuarios restaurar versiones anteriores de un documento, de modo que si se detecta un error o se requiere volver a un estado previo, se pueda habilitar nuevamente esa versión.

Prioridad: Media.

Criterio de aceptación: El usuario selecciona una versión previa de un documento y el sistema la habilita nuevamente, manteniendo a su vez el historial de cambios.



Figura 10: Caso de uso RF-08: Restauración de versiones de documentos

■ RF-09: Búsqueda de Documentos

Este caso de uso corresponde al requisito funcional **RF-09** (descrito en la Tabla 7). El sistema debe permitir a los usuarios realizar búsquedas de documentos empleando distintos criterios como docente, curso o fecha, con la finalidad de facilitar la localización de la información.

Prioridad: Alta.

Criterio de aceptación: La búsqueda devuelve resultados tanto parciales como exactos según el criterio aplicado.



Figura 11: Caso de uso RF-09: Búsqueda de documentos

■ RF-10: Etiquetado de Documentos

Este caso de uso corresponde al requisito funcional **RF-10** (descrito en la Tabla 7). El sistema debe permitir a los usuarios etiquetar documentos con categorías personalizadas, de manera que puedan organizar y clasificar la información de forma más flexible.

Prioridad: Media.

Criterio de aceptación: El usuario puede añadir, editar o eliminar etiquetas y utilizarlas para clasificar documentos.

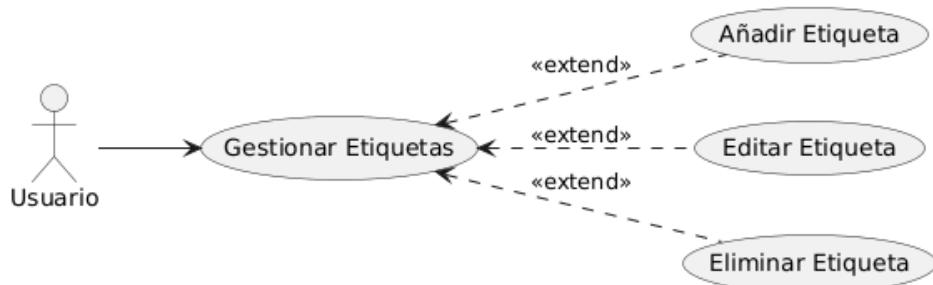


Figura 12: Caso de uso RF-10: Etiquetado de documentos

■ RF-11: Descarga Segura de Documentos

Este caso de uso corresponde al requisito funcional **RF-11** (descrito en la Tabla 7). El sistema debe permitir la descarga de documentos únicamente a los usuarios autorizados, realizando una verificación previa de permisos para garantizar la seguridad en el acceso.

Prioridad: Alta.

Criterio de aceptación: Solo los usuarios autorizados pueden descargar archivos, y el sistema bloquea los intentos de acceso no autorizado.



Figura 13: Caso de uso RF-11: Descarga segura de documentos

■ **RF-12: Gestión Documental - Eliminación Lógica de Documentos**

Este caso de uso corresponde al requisito funcional **RF-12** (descrito en la Tabla 7). El sistema debe permitir la eliminación lógica de documentos, evitando su eliminación permanente en la base de datos y permitiendo su recuperación dentro de un periodo definido.

Prioridad: Media.

Criterio de aceptación: Un documento eliminado deja de mostrarse en la lista activa, pero puede recuperarse dentro del periodo establecido antes de su eliminación definitiva.

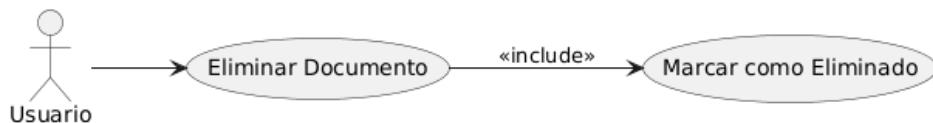


Figura 14: Caso de uso RF-12: Eliminación lógica de documentos

■ **RF-13: Gestión Académica - Administración de Áreas, Grados y Secciones**

Este caso de uso corresponde al requisito funcional **RF-13** (descrito en la Tabla 7). El sistema debe permitir al administrador crear, actualizar y eliminar áreas, grados y secciones, registrando y gestionando sus datos básicos, así como asignándoles un código único para su identificación.

Prioridad: Alta.

Criterio de aceptación: El administrador puede crear, actualizar y eliminar áreas, grados y secciones, y los cambios se reflejan correctamente en el sistema con sus códigos únicos asignados.

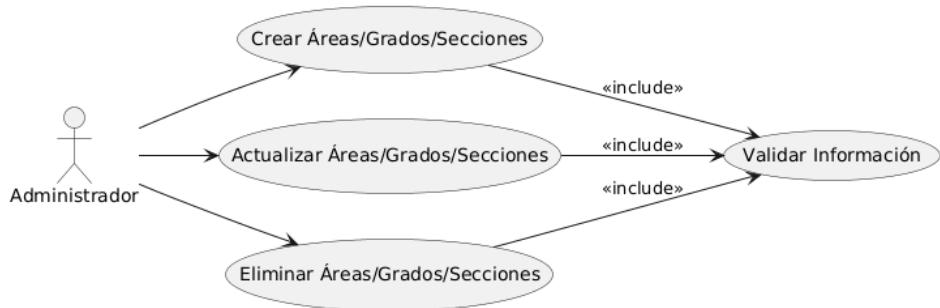


Figura 15: Caso de uso RF-13: Administración de áreas, grados y secciones

■ RF-14: Gestión Académica - Administración de Perfiles de Docentes

Este caso de uso corresponde al requisito funcional **RF-14** (descrito en la Tabla 8). El sistema debe permitir al administrador crear y editar perfiles de docentes, registrando información como sus áreas de especialización y la carga horaria correspondiente.

Prioridad: Alta.

Criterio de aceptación: El administrador puede crear o editar el perfil de un docente, y este se registra correctamente en el sistema siendo visible en las asignaciones.

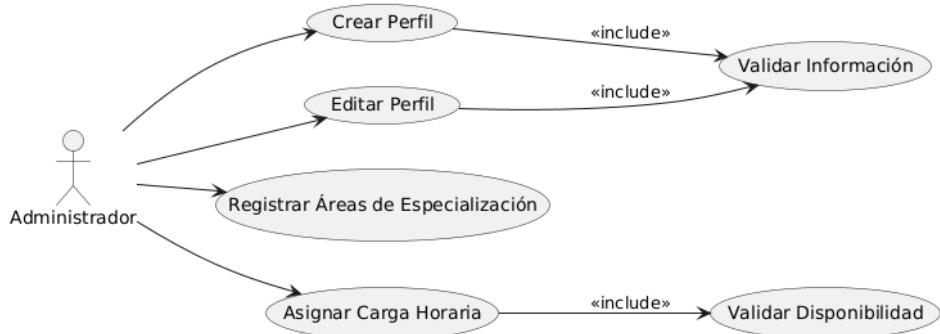


Figura 16: Caso de uso RF-14: Administración de perfiles de docentes

■ RF-15: Planificación de Horarios - Visualización de Calendario

Este caso de uso corresponde al requisito funcional **RF-15** (descrito en la Tabla 8). El sistema debe mostrar un calendario visual y responsive para la planificación de horarios, el cual debe adaptarse a distintos dispositivos como computadoras de escritorio y móviles.

Prioridad: Alta.

Criterio de aceptación: El calendario se visualiza correctamente y se adapta de manera responsive tanto en escritorio como en dispositivos móviles.

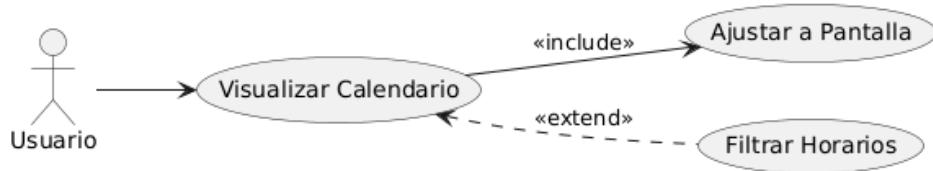


Figura 17: Caso de uso RF-15: Visualización de calendario responsive

- **RF-16: Planificación Horarios - Visualización según rol**

Este caso de uso corresponde al requisito funcional **RF-16** (descrito en la Tabla 8). El sistema debe permitir la visualización de calendarios diferenciados de acuerdo al rol del usuario: los docentes únicamente podrán visualizar sus propios horarios, mientras que los administradores tendrán acceso a los calendarios de todos los docentes.

Prioridad: Alta.

Criterio de aceptación: Un docente solo visualiza su propio calendario y el administrador puede visualizar los calendarios de todos los docentes.

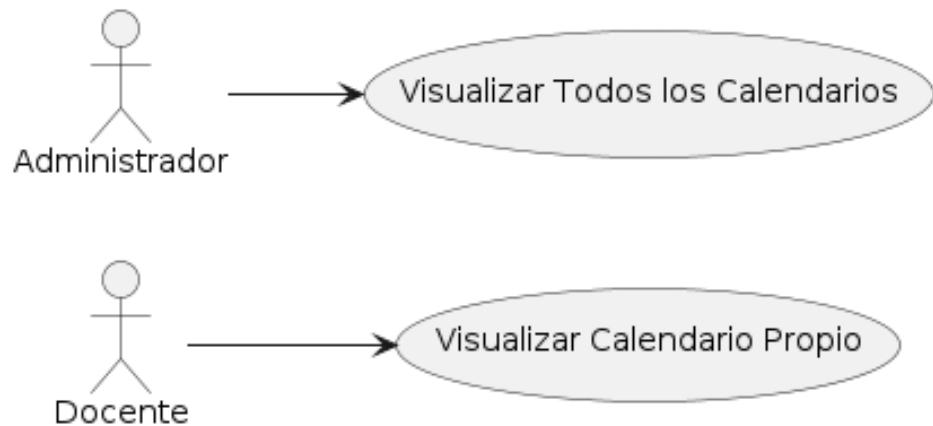


Figura 18: Caso de uso RF-16: Visualización de calendario según rol

- **RF-17: Planificación Horarios - Creación y edición manual con Undo/Redo**

Este caso de uso corresponde al requisito funcional **RF-17** (descrito en la Tabla 8). El sistema debe permitir al usuario crear, editar y eliminar horarios de manera manual, ofreciendo la posibilidad de deshacer (undo) o rehacer (redo) los cambios realizados.

Prioridad: Alta.

Criterio de aceptación: Los cambios realizados en un horario pueden deshacerse o rehacerse correctamente, garantizando la integridad de la planificación.

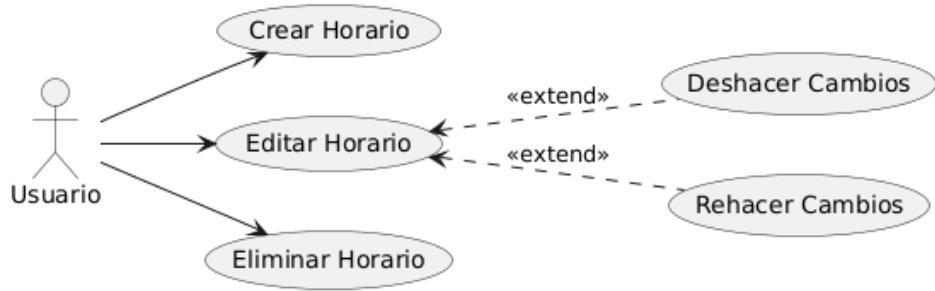


Figura 19: Caso de uso RF-17: Creación y edición manual de horarios con Undo/Redo

■ RF-18: Planificación Horarios - Detección de conflictos

Este caso de uso corresponde al requisito funcional **RF-18** (descrito en la Tabla 8). El sistema debe detectar conflictos en los horarios (por ejemplo, solapamiento de asignaciones) y notificar al usuario cuando se intente guardar un horario que genere choques.

Prioridad: Alta.

Criterio de aceptación: Al guardar un horario que presenta conflictos, el sistema muestra una notificación informando al usuario del choque y evitando la asignación incorrecta.

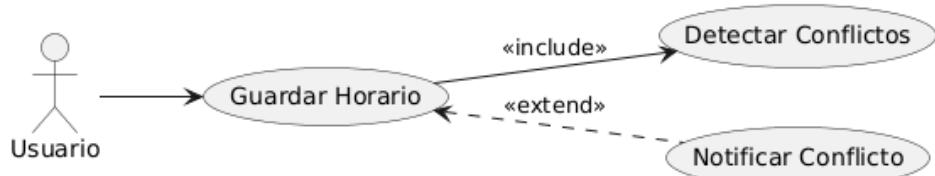


Figura 20: Caso de uso RF-18: Detección de conflictos en horarios

■ RF-19: Asignación Automática - Docentes a áreas

Este caso de uso corresponde al requisito funcional **RF-19** (descrito en la Tabla 8). El sistema debe permitir asignar docentes a áreas de manera automática mediante un algoritmo, teniendo en cuenta criterios como disponibilidad, carga horaria y competencias.

Prioridad: Alta.

Criterio de aceptación: El sistema genera las asignaciones automáticas de docentes a áreas.

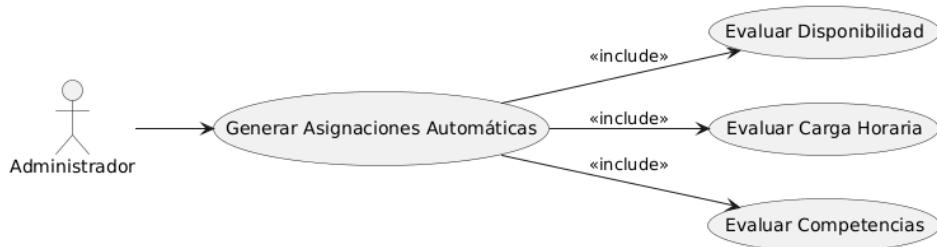


Figura 21: Caso de uso RF-19: Asignación automática de docentes a áreas

- **RF-20: Asignación Automática - Aceptación y ajuste manual**

Este caso de uso corresponde al requisito funcional **RF-20** (descrito en la Tabla 8). El sistema debe permitir que el usuario (coordinador) acepte o rechace las asignaciones automáticas generadas por el sistema, y realice modificaciones manuales en caso de ser necesario.

Prioridad: Media.

Criterio de aceptación: El sistema registra la decisión final del coordinador, incluyendo las modificaciones manuales realizadas sobre las asignaciones automáticas.

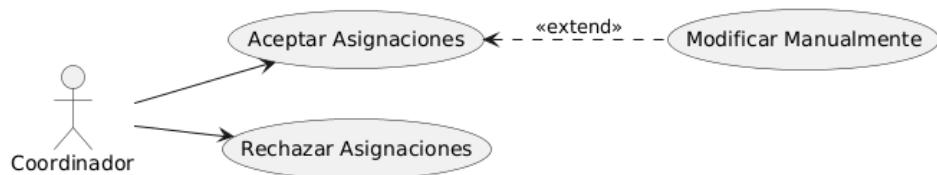


Figura 22: Caso de uso RF-20: Aceptación y ajuste manual de asignaciones automáticas

- **RF-21: Exportación de Horarios - PDF y CSV**

Este caso de uso corresponde al requisito funcional **RF-21** (descrito en la Tabla 8). El sistema debe permitir exportar los horarios generados a archivos PDF y CSV, siguiendo la plantilla institucional para mantener consistencia y formato estandarizado.

Prioridad: Media.

Criterio de aceptación: Los archivos generados cumplen con el formato institucional y se descargan correctamente desde el sistema.

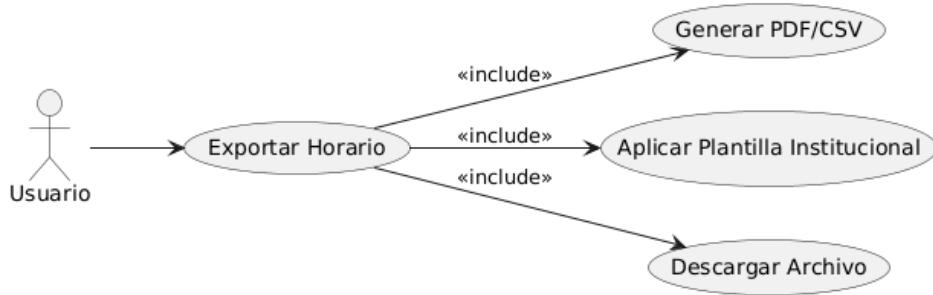


Figura 23: Caso de uso RF-21: Exportación de horarios a PDF y CSV

■ **RF-22: Seguridad - Backup automático mensual**

Este caso de uso corresponde al requisito funcional **RF-22** (descrito en la Tabla 8). El sistema debe generar un backup automático de los datos de manera mensual, conservando únicamente la copia más reciente para optimizar espacio de almacenamiento.

Prioridad: Media.

Criterio de aceptación: El backup se ejecuta en la fecha configurada y puede ser recuperado correctamente cuando se necesite.

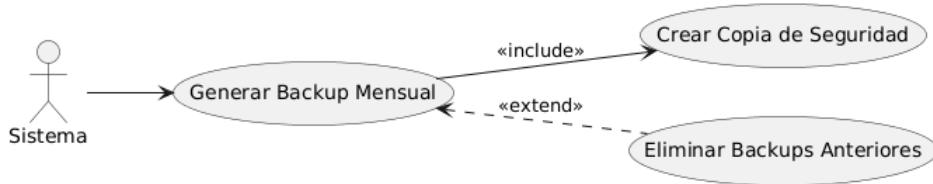


Figura 24: Caso de uso RF-22: Backup automático mensual

■ **RF-23: Seguridad - Control de accesos mediante ACL**

Este caso de uso corresponde al requisito funcional **RF-23** (descrito en la Tabla 8). El sistema debe permitir el control de acceso a documentos mediante listas de control de acceso (ACL), asignando permisos específicos por usuario o por rol.

Prioridad: Alta.

Criterio de aceptación: Los permisos de acceso se aplican correctamente en cada intento de lectura o descarga, respetando las restricciones definidas por usuario o rol.

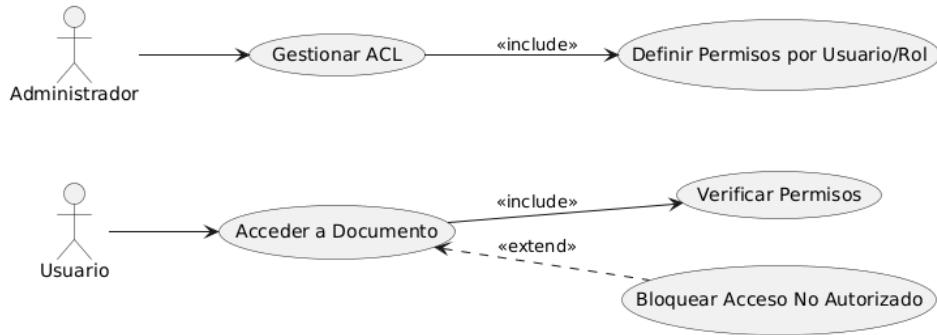


Figura 25: Caso de uso RF-23: Control de accesos mediante ACL

3.2.3. Requerimientos No Funcionales

- **RNF-01 Rendimiento:** Las búsquedas deben responder en menos de 4 segundos para conjuntos de hasta 10,000 documentos.
- **RNF-02 Disponibilidad:** El sistema debe estar disponible al menos el 95 % del tiempo durante horario laboral.
- **RNF-03 Escalabilidad:** La arquitectura debe soportar un crecimiento del 100 % en usuarios activos sin reestructuración mayor.
- **RNF-04 Seguridad:** Las contraseñas deben almacenarse cifradas con algoritmo de hashing.
- **RNF-05 Compatibilidad:** El sistema debe ser accesible desde navegadores modernos (Chrome, Firefox, Edge).

3.2.4. Suposiciones y Restricciones

- **Restricción 1:** El sistema se desplegará en una red LAN institucional, con acceso restringido desde Internet.
- **Restricción 2:** Solo se soportarán formatos de documentos PDF, DOCX y ZIP.
- **Restricción 3:** El sistema dependerá de un servicio SMTP institucional para el envío de correos de recuperación de contraseña.
- **Suposición 1:** Cada docente posee una cuenta institucional válida para autenticación.
- **Suposición 2:** La base de datos inicial de áreas y docentes será proporcionada por la institución.

El modelo de clases UML 26 del sistema integrado de gestión documental y administrativa ha sido diseñado siguiendo los principios de orientación a objetos. La arquitectura del modelo

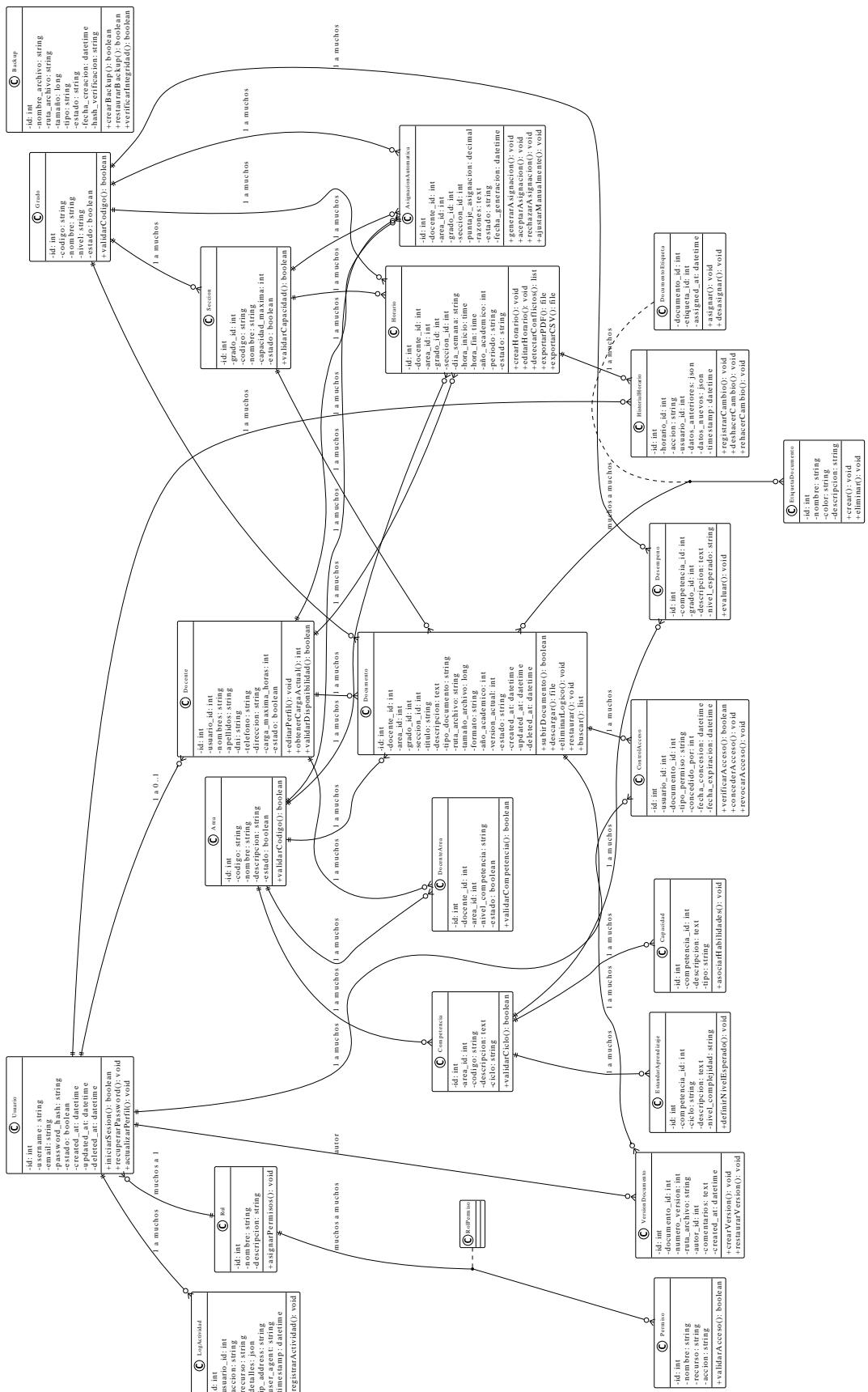


Figura 26: Modelo de Clases UML

refleja la estructura organizacional de la Institución Educativa Padre Pérez de Guereñu y se alinea con los conceptos fundamentales del Marco Curricular Nacional del Perú.

El dominio de gestión de usuarios comprende las clases Usuario, Rol, Permiso y Docente. Este dominio implementa el sistema de autenticación, autorización y gestión de perfiles. La clase Usuario actúa como la entidad base para el acceso al sistema, mientras que Docente extiende la información específica del personal académico. La relación entre Rol y Permiso permite una gestión flexible de accesos mediante listas de control (ACL).

En la gestión académica se incluye las clases Area, Grado, Seccion y DocenteArea. Este dominio modela la estructura académica institucional, permitiendo la administración de las áreas de conocimiento, niveles educativos y la relación de competencias entre docentes y áreas específicas. Junto con ello se agrega, Competencia, Capacidad, Desempeno y EstandarAprendizaje. Estas clases implementan directamente los conceptos definidos en el MCN peruano, garantizando que la arquitectura de información del sistema esté alineada con las políticas educativas nacionales. Centrandonos en la gestión documental tenemos las clases Documento, VersionDocumento, EtiquetaDocumento y DocumentoEtiqueta. Este dominio maneja el ciclo de vida completo de los documentos pedagógicos, incluyendo versionado, clasificación por etiquetas y eliminación lógica. La planificación de horarios incluye, Horario, AsignacionAutomatica y HistorialHorario. Este dominio implementa tanto la gestión manual como automática de horarios, incluyendo detección de conflictos. Para el domingio de la seguridad se implemento los modelos ControlAcceso, LogActividad y Backup. Este dominio garantiza la seguridad del sistema mediante el control granular de accesos, auditoría de actividades y estrategias de respaldo.

3.2.5. Patrones de Diseño Implementados

El modelo incorpora varios patrones de diseño reconocidos: Patrón Command: Reflejado en HistorialHorario para soportar las funcionalidades de undo/redo en la planificación de horarios. Patrón Soft Delete: Aplicado en las clases principales (Documento, Horario) mediante el atributo deleted_at, permitiendo recuperación de datos eliminados accidentalmente.

3.2.6. Consideraciones de Integridad y Consistencia

El modelo garantiza la integridad referencial mediante relaciones bien definidas. Las cardinalidades están diseñadas para reflejar las restricciones del negocio: un docente puede estar asignado a múltiples áreas pero con limitaciones de carga horaria, un documento pertenece a un único docente autor pero puede tener múltiples versiones, y un horario debe validar la no existencia de conflictos temporales. La implementación del versionado en documentos mantiene la trazabilidad completa de cambios, permitiendo auditorías y recuperación de versiones anteriores sin pérdida de información histórica. Este modelo de clases constituye la base arquitectónica que soporta todos los requerimientos funcionales identificados en el sistema, proporcionando flexibilidad para futuras extensiones manteniendo la cohesión y bajo acoplamiento entre los diferentes dominios del negocio.

3.3. Diagrama de Secuencia UML

■ DS-03: Gestión de Usuarios

El diagrama de secuencia **DS-03** nace del requerimiento funcional **RF-01**. Este diagrama **DS-03** modela el proceso de autenticación que permite a los usuarios acceder al sistema.

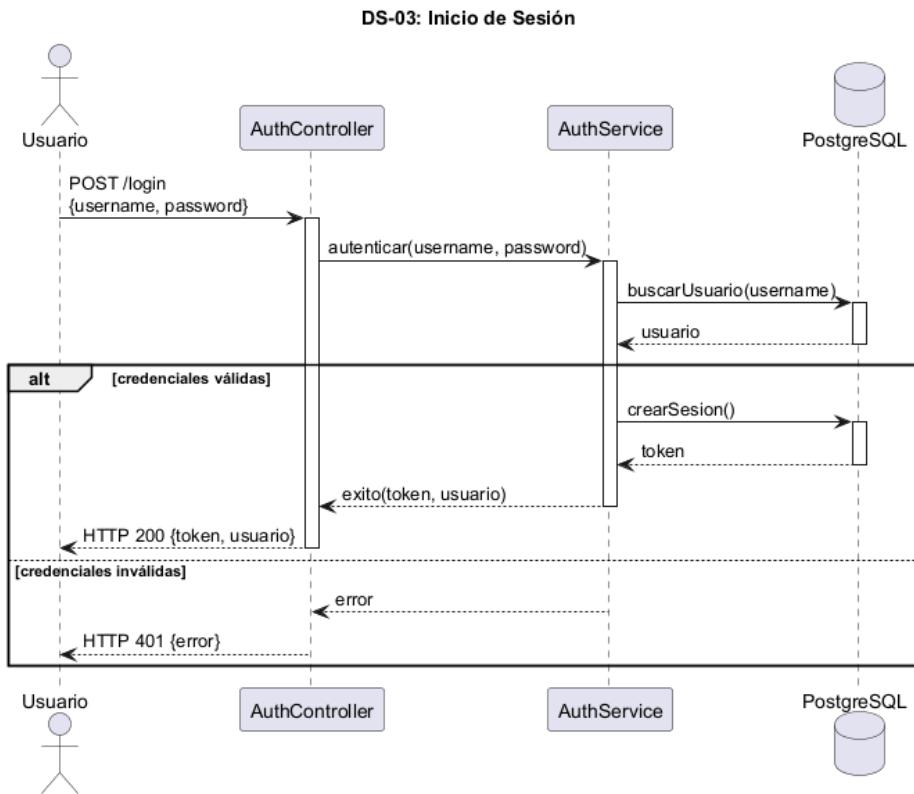


Figura 27: Diagrama de secuencia DS-03: Detección de Conflictos en Horarios

Este diagrama **DS-04** modela el proceso de creación de nuevas cuentas de usuario por parte de administradores autorizados.

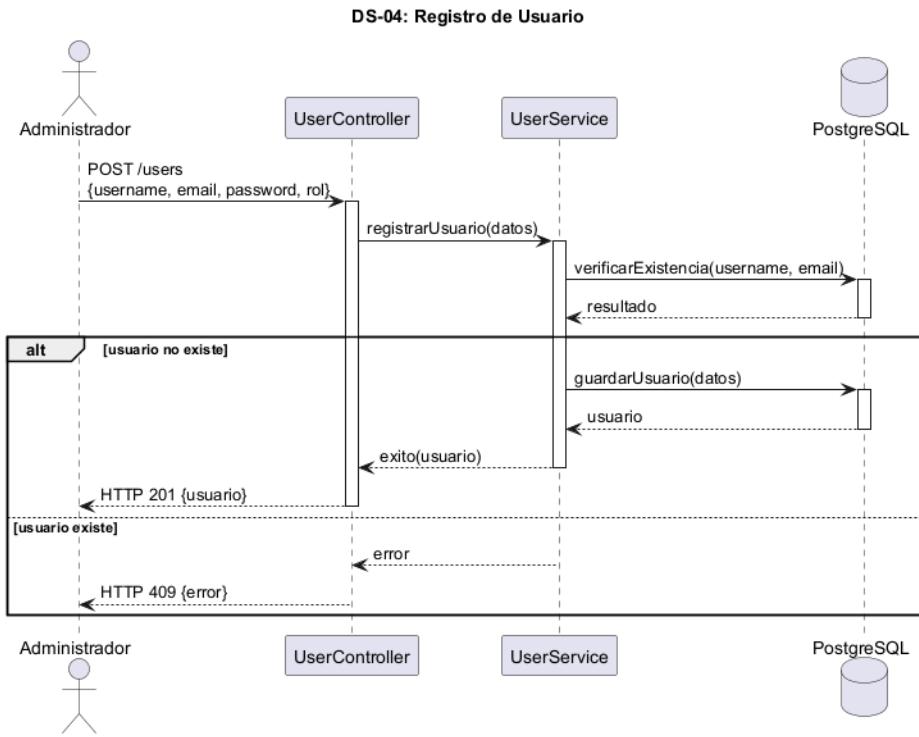


Figura 28: Diagrama de secuencia DS-04: Detección de Conflictos en Horarios

■ DS-01 - Asignación Automática de Docentes

El diagrama de secuencia **DS-01** nace del requerimiento funcional **RF-19**. **DS-01** representa la funcionalidad más compleja y diferenciadora del sistema propuesto. Este requerimiento automatiza el proceso tradicionalmente manual de asignación académica.

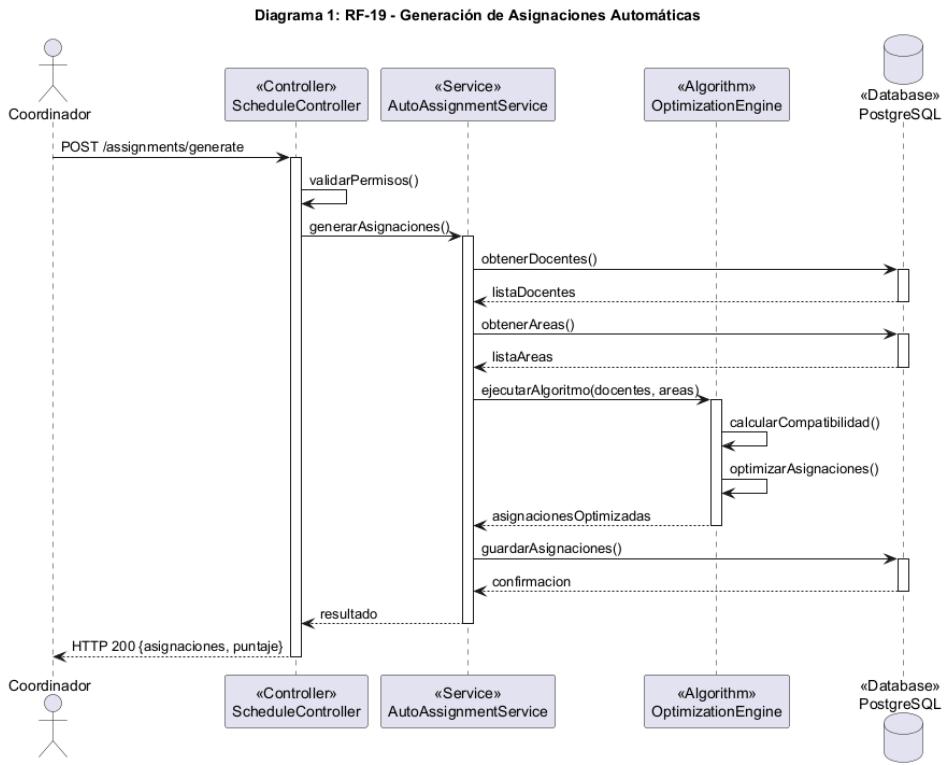


Figura 29: Diagrama de secuencia DS-01: Asignación Automática de Docentes

■ DS-02: Detección de Conflictos en Horarios

El diagrama de secuencia DS-02 nace del requerimiento funcional RF-18. El DS-02 muestra un mecanismo crítico de validación que previene la creación de horarios con solapamientos temporales. Este requisito garantiza la integridad de los datos académicos y evita asignaciones imposibles de cumplir físicamente.

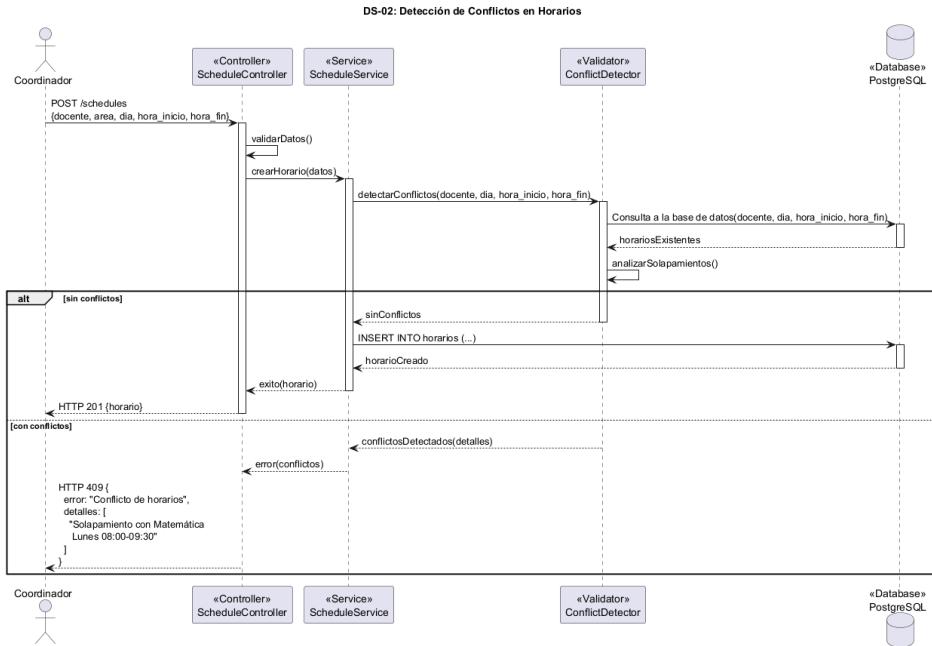


Figura 30: Diagrama de secuencia DS-02: Detección de Conflictos en Horarios

3.4. Arquitectura del Sistema

La arquitectura del sistema será representada por un diagrama de componentes UML. La arquitectura del sistema se plantea bajo un enfoque multicapa, organizada en tres niveles principales: presentación, aplicación y datos.

- En la capa de presentación los principales usuarios del sistema son docentes, administrativos y personal académico. La interacción con el sistema se realiza a través de un navegador web, garantizando acceso multiplataforma (computadoras y dispositivos móviles). La interfaz es responsive y está diseñada para simplificar la interacción con los módulos académico y administrativo.

- En la capa de aplicación tenemos el servidor Laravel (sobre Apache) donde se encuentran el módulo académico y Administrativo

Servidor Laravel (sobre Apache): Este servidor ejecuta la lógica del negocio y organiza el flujo entre los módulos. Se optó por Laravel debido a su robustez, seguridad y soporte en la comunidad de software libre.

Módulos Principales:

Módulo Académico: Gestiona documentos pedagógicos (planificaciones, sesiones, mate-

riales de clase). Implementa almacenamiento seguro, versionado y motor de búsqueda, reduciendo la duplicidad y pérdida de información.

Módulo Administrativo: Centraliza la planificación y asignación de horarios, así como la gestión de docentes, áreas y secciones. Incluye un algoritmo propio para la asignación automática de carga académica optimizada.

Servicios de Apoyo:

Cache: Mejora el rendimiento de las consultas.

Storage: Maneja almacenamiento de archivos.

Seguridad (Sesiones): Controla autenticación, roles y permisos de usuarios.

■ Capa de Datos

Servidor PostgreSQL: Base de datos principal donde se almacenan usuarios, roles, datos académicos, documentos, logs y metadatos.

Servicios de Seguridad y Mantenimiento:

Backups automáticos: Configurados vía cronjobs con historial mensual para recuperación ante fallos.

Monitoreo (PgAdmin y logs): Permite auditoría de actividad.

Almacenamiento seguro: Garantiza integridad de la información.

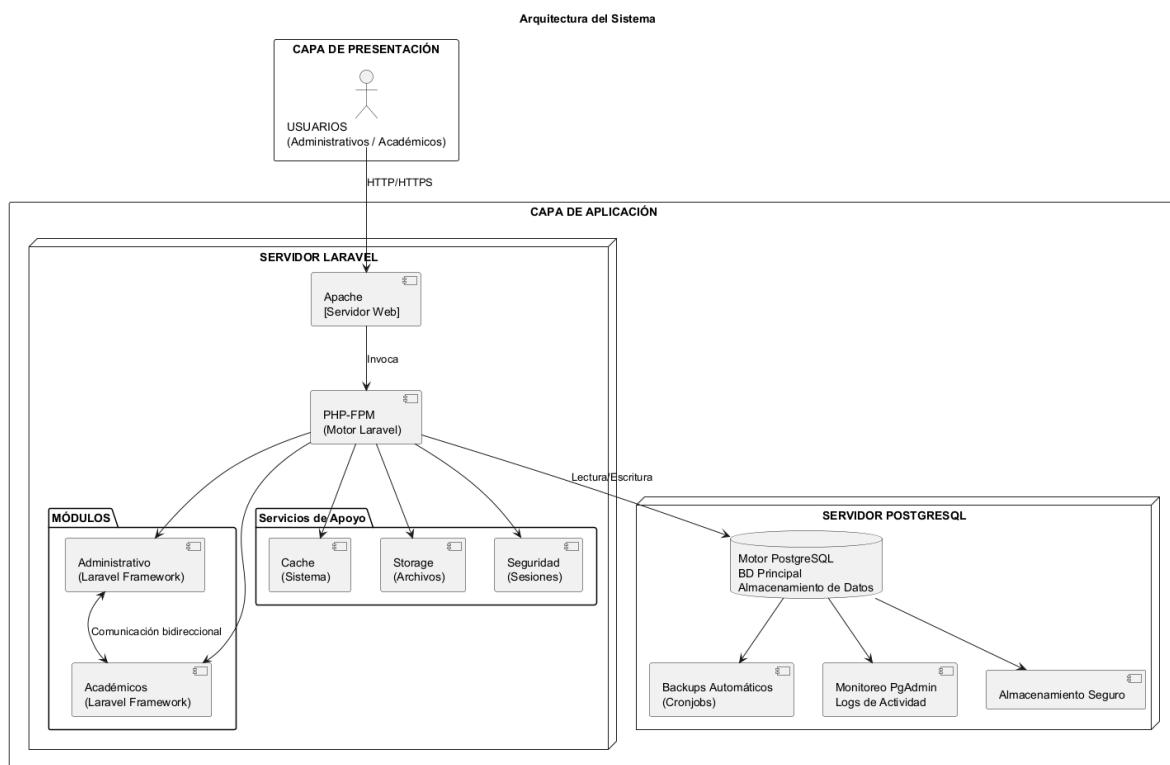


Figura 31: Diagrama de la arquitectura del sistema

3.5. Diseño de la Base de Datos

3.5.1. Arquitectura de Base de Datos

El sistema utiliza una arquitectura de base de datos relacional diseñada para gestionar el proceso completo de planificación académica curricular. El diagrama ER 32 muestra las 37 entidades principales y sus relaciones, organizadas en los siguientes módulos funcionales:

3.5.1.1. Módulo de Gestión de Usuarios y Permisos

- `user`: Información básica de usuarios del sistema
- `role`: Roles disponibles en el sistema
- `permission`: Permisos granulares
- `user_role`: Relación muchos a muchos entre usuarios y roles
- `role_permission`: Relación muchos a muchos entre roles y permisos
- `teacher`: Perfil específico de docentes (extiende `user`)
- `student`: Perfil específico de estudiantes (extiende `user`)

3.5.1.2. Módulo de Estructura Curricular

- `year_scholar`: Año escolar
- `diversification`: Diversificación curricular por docente y año
- `diversification_section_grade`: Asignación de diversificaciones a secciones/grados
- `competence`: Competencias del currículo nacional
- `capacity`: Capacidades derivadas de competencias
- `achievement`: Desempeños/logros esperados
- `methodological_guideline`: Orientaciones metodológicas
- `evaluation_guideline`: Orientaciones de evaluación
- `evaluation`: Tipos de evaluación

3.5.1.3. Módulo de Planificación Anual

- `annual_planning`: Planificación anual por diversificación
- `learning_context`: Contextos de aprendizaje

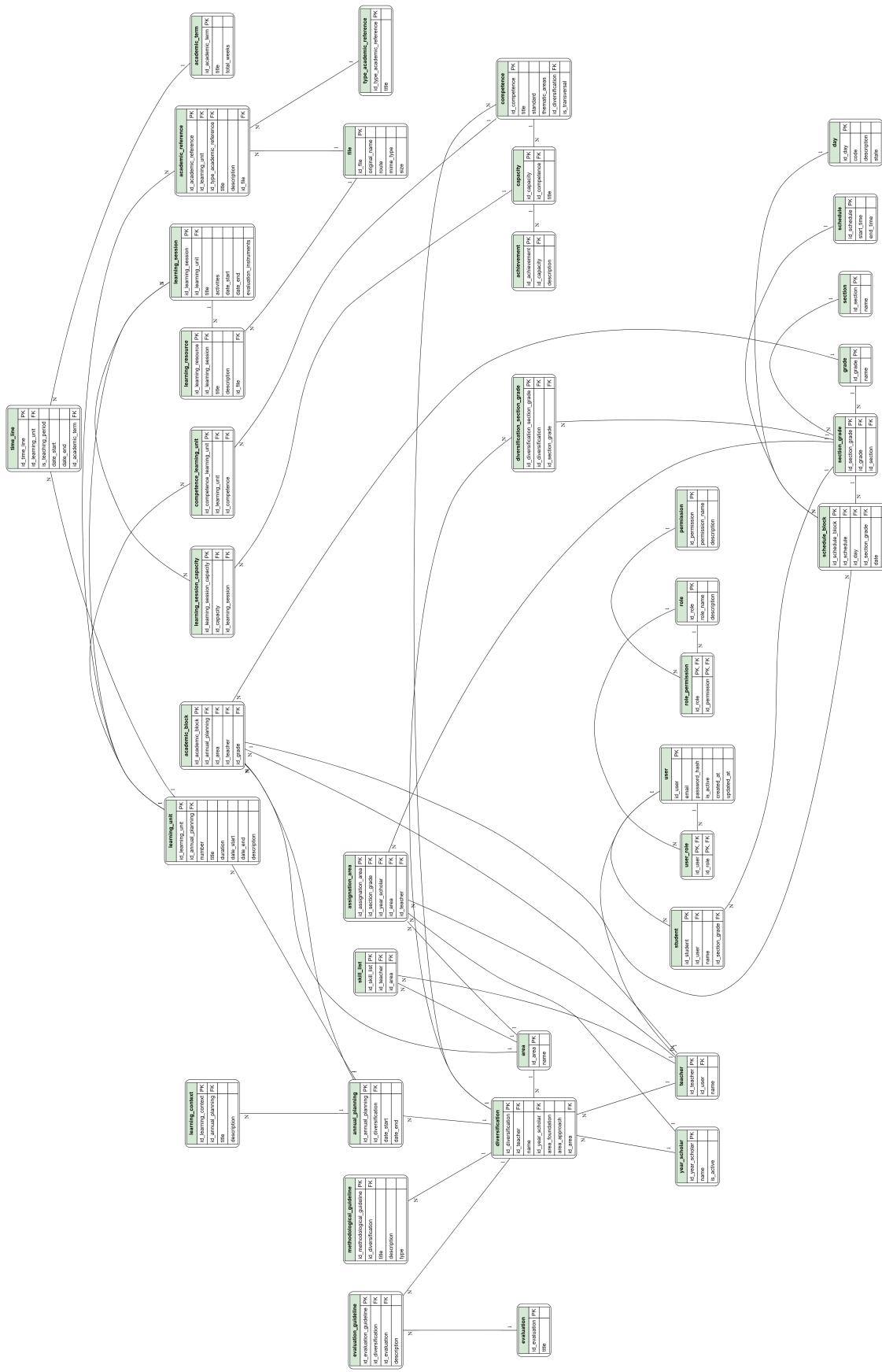


Figura 32: Diagrama de entidad-relación de la base de datos

- `learning_unit`: Unidades didácticas
- `competence_learning_unit`: Competencias trabajadas por unidad
- `academic_term`: Bimestres/trimestres
- `time_line`: Cronograma de unidades
- `learning_session`: Sesiones de aprendizaje
- `learning_session_capacity`: Capacidades por sesión
- `learning_resource`: Recursos educativos
- `academic_reference`: Referencias bibliográficas
- `type_academic_reference`: Clasificación de referencias
- `file`: Gestión de archivos adjuntos

3.5.1.4. Módulo de Organización Escolar

- `grade`: Grados académicos
- `section`: Secciones
- `section_grade`: Relación grado-sección
- `area`: Áreas curriculares
- `skill_list`: Lista de habilidades por área y docente
- `academic_block`: Bloques académicos (asignación docente-área-grado)
- `schedule_block`: Bloques de horario
- `schedule`: Horarios
- `day`: Días de la semana

3.5.2. Normalización

El esquema de base de datos ha sido diseñado siguiendo las formas normales hasta la Tercera Forma Normal (3FN), asegurando:

3.5.2.1. Primera Forma Normal (1FN)

- Todos los atributos contienen valores atómicos
- No existen grupos repetitivos
- Cada tabla tiene una clave primaria definida (UUID)

3.5.2.2. Segunda Forma Normal (2FN)

- Cumple con 1FN
- Todos los atributos no clave dependen completamente de la clave primaria
- Se eliminaron dependencias parciales mediante la creación de tablas intermedias como `user_role, role_permission, competence_learning_unit, learning_session_capacity`

3.5.2.3. Tercera Forma Normal (3FN)

- Cumple con 2FN
- No existen dependencias transitivas
- Ejemplo: La información del usuario está separada de los perfiles específicos (`teacher, student`), evitando redundancia

3.5.2.4. Consideraciones de Desnormalización Controlada

En algunos casos específicos se ha optado por una desnormalización controlada por razones de rendimiento:

- `learning_unit.duration`: Se almacena directamente aunque podría calcularse desde `time_line`, para optimizar consultas frecuentes
- `student.name` y `teacher.name`: Se duplican desde potenciales tablas de perfil para reducir joins en consultas de listado

Estas decisiones se documentan y justifican por el balance entre integridad y rendimiento en operaciones críticas del sistema.

3.6. Implementación del Sistema

3.6.1. Tecnologías Utilizadas

La selección del stack tecnológico para el sistema integrado de gestión documental y administrativa se realizó considerando criterios de madurez tecnológica, compatibilidad con la infraestructura existente, experiencia del equipo de desarrollo y disponibilidad de soporte comunitario. La arquitectura adoptada prioriza el uso de tecnologías de código abierto para eliminar costos de licenciamiento y garantizar la sostenibilidad a largo plazo del proyecto. Es-

te enfoque cobra aún mayor relevancia si se considera que en [33], aproximadamente el 73 % de los participantes de su estudio identificaron que el costo de las licencias de tecnologías o software digital como una barrera alta o suficiente para su uso en entornos educativos , lo que refuerza el optar por soluciones libres y de bajo costo.

3.6.1.1. Stack Tecnológico Principal

Backend y Lógica de Negocio

Laravel Framework v11.45.1 constituye el núcleo del sistema, ejecutándose sobre PHP 8.2.12. La elección de Laravel se fundamenta en su arquitectura MVC robusta que facilita la separación de responsabilidades, su ecosistema maduro de paquetes, y las características de seguridad integradas como protección CSRF, prevención de inyección SQL mediante Eloquent ORM, y sistema de autenticación robusto.

El servidor web Apache HTTP Server v2.4.57 fue seleccionado debido a su compatibilidad comprobada con el servidor local proporcionado por la institución educativa.

Sistema de Gestión de Base de Datos

PostgreSQL v15.3 fue seleccionado como sistema gestor de base de datos relacional por su superioridad en el manejo de transacciones ACID, su rendimiento en consultas complejas con múltiples joins fundamentales para el funcionamiento del algoritmo de asignación automática de docentes.

La interfaz de usuario se desarrolló utilizando estándares web modernos (HTML5, CSS3, JavaScript ES6+) con Bootstrap 5.3.0 como framework CSS principal. La elección de Bootstrap se fundamenta en su sistema de grid responsive probado, componentes preconstruidos, y amplia documentación que acelera el desarrollo y garantiza consistencia visual across dispositivos.

Para la funcionalidad de calendario y gestión de horarios, se desarrolló un componente personalizado utilizando Bootstrap, optimizado específicamente para los requerimientos de la institución educativa.

Tabla 9: Stack Tecnológico del Sistema

Tecnología	Versión	Capa	Justificación de Selección
Laravel	11.45.1	Backend	Seguridad integrada, ORM robusto, soporte LTS, experiencia del equipo
PHP	8.2.12	Backend	Compatibilidad con Laravel 11, mejoras de rendimiento, tipado estricto
Apache	2.4.57	Servidor	Disponibilidad en servidor institucional, madurez comprobada
PostgreSQL	15.3	Base de Datos	Rendimiento en queries complejas, tipos avanzados, full-text search
Bootstrap	5.3.0	Frontend	Sistema responsive, componentes preconstruidos, documentación extensa

3.6.1.2. Herramientas de Desarrollo y DevOps Control de Versiones y Gestión de Dependencias

El proyecto se gestionó mediante Git v2.42.0 con repositorio en GitHub. Composer v2.6.5 gestiona las dependencias PHP con versiones bloqueadas mediante composer.lock para reproducibilidad, mientras que npm v10.2.3 maneja los paquetes JavaScript.

Entorno de Desarrollo Containerizado

Se implementó Docker v24.0.7 con Docker Compose para containerización del entorno de desarrollo, garantizando consistencia entre los equipos de los testistas.

3.6.2. Desarrollo de los Módulos

El sistema se estructuró en dos módulos principales con responsabilidades específicas: el Módulo Académico para gestión documental pedagógica, y el Módulo Administrativo para planificación de horarios y asignación de carga académica. Ambos módulos comparten servicios de autenticación, autorización, auditoría y logging, implementados siguiendo el principio de responsabilidad única asegurándose que cada clase o módulo debe encargarse de una sola

funcionalidad dentro del sistema.

Para la nomenclatura de base de datos se utilizó snake_case en minúsculas: tablas en plural (annual_plannings, learning_units), columnas descriptivas (created_at, teacher_id), claves foráneas con sufijo _id, y soft deletes implementados mediante columna nullable deleted_at.

La seguridad se implementó transversalmente mediante sanitización automática de entradas con htmlspecialchars() y validadores Laravel que filtran caracteres peligrosos, consultas preparadas exclusivamente a través de Eloquent ORM previniendo inyección SQL, tokens CSRF obligatorios en todos los formularios, y verificación de permisos en cada método de controlador y Policies de Laravel.

3.6.2.1. Módulo Académico El Módulo Académico implementa la gestión completa del ciclo de vida de documentos pedagógicos, abarcando las funcionalidades de almacenamiento, versionado, búsqueda, clasificación, control de accesos y eliminación lógica definidas en los requerimientos **RF-04 a RF-12 y RF-22, RF-23**.

- **Arquitectura y Componentes.** El módulo sigue una arquitectura en capas claramente diferenciadas: Controladores (capa HTTP que recibe requests y devuelve responses), Form Requests (capa de validación que centraliza reglas y mensajes), Servicios (capa de lógica de negocio donde residen los algoritmos de la lógica del negocio), y Modelos Eloquent (capa de persistencia que representa entidades de base de datos).
- **Sistema de Almacenamiento de Documentos.** El almacenamiento de documentos implementa una estructura jerárquica estandarizada en el filesystem:
`storage/app/documents/{año_escolar}/{teacher_id}/{tipo_documento}/`,
donde cada documento se almacena con nombre único generado mediante hash SHA-256 del contenido más timestamp, previniendo colisiones y facilitando detección de duplicados. El proceso de subida valida formato de archivo (PDF, DOCX, ZIP), tamaño máximo (10 MB), y MIME type real (no solo extensión, previniendo ataques por archivo malicioso renombrado). El almacenamiento se realiza dentro de una transacción de base de datos que crea simultáneamente el registro en la tabla documents con todos los metadatos (teacher_id, area_id, grade_id, section_id, título, descripción, tipo, ruta, tamaño, formato, año académico, versión actual, estado) y registra la acción en la auditoría. Si cualquier paso falla, la transacción se revierte automáticamente garantizando consistencia.
- **Sistema de Versionado.**

El versionado permite mantener un historial completo de modificaciones en documentos. Cada vez que un usuario sube una nueva versión de un documento existente, el sistema

incrementa el contador de versión (únicamente hasta 2 versiones), almacena el nuevo archivo en el filesystem con nomenclatura versionada, crea un registro en la tabla `document_versions` con número de versión, ruta del archivo, `author_id` (quien realizó el cambio), comentarios opcionales, hash SHA-256 para verificación de integridad, y actualiza el documento principal para apuntar a la nueva versión como `versión_actual`.

- **Motor de Búsqueda.**

El motor de búsqueda implementa filtros múltiples combinables: por docente (`teacher_id`), área curricular (`area_id`), grado académico (`grade_id`), año escolar (`academic_year`), rango de fechas (`created_at between`).

- **Control de Accesos Granular.**

El sistema de control de accesos opera bajo un modelo híbrido que combina Permisos Basados en Roles (RBAC) para funciones generales del sistema, y Permisos Basados en Propiedad para la gestión de documentos. El autor de un documento siempre goza de acceso completo a su creación. El acceso a los documentos de otros propietarios está reservado exclusivamente a los administradores del sistema.

3.6.2.2. Módulo Administrativo El Módulo Administrativo constituye el núcleo diferenciador del sistema, implementando funcionalidades avanzadas de planificación de horarios con asignación automática de docentes mediante algoritmo propio, detección de conflictos en tiempo real, gestión de carga académica e interfaz visual interactiva tipo calendario.

- **Arquitectura y Componentes.** Los componentes principales son: `ScheduleService`, que gestiona el CRUD de horarios con validaciones de carga horaria y disponibilidad; `ConflictDetectionService`, que implementa el algoritmo de detección de conflictos evaluando solapamientos de docentes, aulas, límites de carga diaria y períodos sin descanso; `AutomaticAssignmentService`, que ejecuta el algoritmo propio de asignación automática considerando competencias del docente en áreas, disponibilidad horaria, balanceo de carga y preferencias configuradas; `ScheduleHistoryService`, que implementa el patrón *Command* para registro de cambios con soporte de *undo/redo*; y `ScheduleExportService`, que genera archivos PDF y CSV con el formato institucional para impresión y distribución.
- **Sistema de Gestión de Horarios.** Los horarios se modelan en la tabla `schedules` con campos: `teacher_id`, `area_id`, `grade_id`, `section_id` (definiendo qué docente enseña qué área a qué grado y sección), `day_of_week` (enum: 'monday' a 'friday'), `start_time` y `end_time` (tipo `time`), `academic_year`, `period` (bimestre/trimestre) y `status` ('active', 'pending', 'deleted').

La creación de un horario ejecuta una secuencia de validaciones:

1. Verificar que el docente tenga competencia en el área asignada consultando la tabla `docente_area`.
2. Calcular las horas semanales actuales del docente sumando la duración de todos sus bloques activos.
3. Verificar que añadir el nuevo bloque no exceda la carga máxima semanal definida en el perfil del docente.
4. Ejecutar el algoritmo de detección de conflictos.
5. Si se detectan conflictos de severidad alta y el parámetro `force_create` no está activado, rechazar la creación y devolver los conflictos al cliente.

Si todas las validaciones pasan, se crea el registro de horario dentro de una transacción de base de datos que también: registra el cambio en `schedule_history` para soporte de `undo/redo`, actualiza el campo `current_weekly_hours` en la tabla `teachers` y registra la acción en `log_activity`. Esta atomicidad garantiza consistencia incluso ante fallos del sistema.

- **Algoritmo de Detección de Conflictos.** El algoritmo evalúa múltiples tipos de incompatibilidades con severidades diferenciadas.

Conflictos de Severidad Alta (bloquean la operación):

- **Solapamiento de docente:** el mismo docente está asignado a dos clases simultáneas. Se detecta buscando horarios existentes del docente en el mismo `day_of_week` donde el rango `[start_time, end_time]` del nuevo bloque se solapa con bloques existentes. Un solapamiento existe si:

`(nuevo_inicio < existente_fin) AND (nuevo_fin > existente_inicio)`.

- **Solapamiento de aula:** la misma sección tiene dos clases simultáneas con diferentes docentes. Se detecta análogamente buscando horarios existentes para la combinación `grade_id + section_id` en el mismo día con solapamiento temporal.

Conflictos de Severidad Media (advertencias con override permitido):

- **Incompatibilidad de competencia:** el docente no tiene registrada competencia en el área asignada. Se verifica consultando la tabla `docente_area`.

Conflictos de Severidad Baja (informativos):

- **Límite diario excedido:** el docente superaría las 6 horas pedagógicas en un día (límite recomendado por normativa educativa). Se calcula sumando las duraciones de todos los bloques del docente en ese día.
- **Bloques consecutivos sin descanso:** el docente tendría más de 3 bloques consecutivos sin un intervalo mínimo de 15 minutos. Se detecta ordenando los bloques del docente en el día y calculando los intervalos entre `end_time` de un bloque y `start_time` del siguiente.

El algoritmo devuelve un array de conflictos con estructura: tipo (code), severidad ('high', 'medium', 'low'), mensaje descriptivo y datos del horario conflictivo o situación problemática. La interfaz de usuario presenta estos conflictos: los de severidad alta bloquean la acción y requieren corrección, mientras que los de media y baja permiten proceder previa confirmación.

3.6.2.3. Implementación del Frontend La implementación del frontend se ha centrado en la creación de una interfaz de usuario intuitiva, responsive y eficiente, que facilite la interacción de los docentes y coordinadores con el sistema. Para lograrlo, se utilizó un conjunto de tecnologías web estandarizadas que aseguran la compatibilidad con navegadores modernos y una experiencia de usuario consistente en distintos dispositivos.

La base del frontend es Bootstrap, un framework de CSS que ha permitido desarrollar un diseño adaptable (responsive) de forma ágil, garantizando que la visualización y funcionalidad del sistema sean óptimas tanto en equipos de escritorio como en tabletas y dispositivos móviles. La estructura de las vistas se gestionó a través de Blade, el motor de plantillas de Laravel, que facilita la organización del código HTML en componentes reutilizables y mantenibles.

Se ha hecho un uso extensivo de las características de Blade, como la herencia de plantillas (@extends), las secciones (@section) y la inclusión de componentes (@include), para evitar la duplicidad de código y centralizar elementos comunes de la interfaz, como la barra de navegación, el pie de página y los menús laterales.

```
1
2
3 @extends('layouts.app')
4
5 @section('content')
6 <div class="container">
7   <div class="row justify-content-center">
8     <div class="col-md-8">
9       <div class="row gx-5 align-items-center">
10         <div class="col-xxl-5">
11           <div class="text-center text-xxl-start">
12             <div class="fs-3 fw-light text-muted">Facilita
tu día a día y dedica más tiempo a lo que realmente importa: Educar
sin límites.</div>
13           <h1 class="display-3 fw-bolder mb-5"><span
class="text-gradient d-inline">Yachay</span></h1>
```

```

14             <div class="d-grid gap-3 d-sm-flex
15               justify-content-sm-center justify-content-xxl-start mb-3">
16
17               @if(Auth::user()->roles->contains('role_name', 'administrador'))
18                   <a class="btn btn-dark btn-lg px-5 py-3
19                     me-sm-3 fs-6 fw-bolder" href="{{route('teacher.index')}}">Docentes</a>
20
21                   <a class="btn btn-outline-dark btn-lg px-5
22                     py-3 fs-6 fw-bolder" href="{{route('schedule_block.schedule_view')}}">Horarios</a>
23
24                   @else
25                       <a class="btn btn-dark btn-lg px-5 py-3
26                         me-sm-3 fs-6 fw-bolder" href="{{route('teacher_home.index')}}">Mis
27                         áreas</a>
28
29                         <a class="btn btn-outline-dark btn-lg px-5
30                           py-3 fs-6 fw-bolder" href="{{route('teacher_home.units')}}">Mis
31                           unidades</a>
32
33                   @endif
34               </div>
35
36               </div>
37
38               <div class="col-xxl-7">
39                   <div class="d-flex justify-content-center mt-5
40                     mt-xxl-0">
41
42                       <img src="{{ asset('img/yachay_logo.png') }}>
43                   </div>
44
45               </div>
46
47           </div>
48
49       </div>
50
51   </div>
52
53 @endsection

```

Listing III.1: Fragmento de home.blade.php

Arquitectura de Vistas y Componentes Visuales La arquitectura del frontend sigue el patrón Modelo-Vista-Controlador (MVC) provisto por Laravel. Las vistas, desarrolladas con Blade, se organizan en una estructura modular que corresponde a las distintas funcionalidades del sistema, como la gestión de documentos, la administración de usuarios y la planificación de horarios.

A continuación, se presentan algunas capturas de pantalla que ilustran la implementación de los componentes visuales más relevantes.

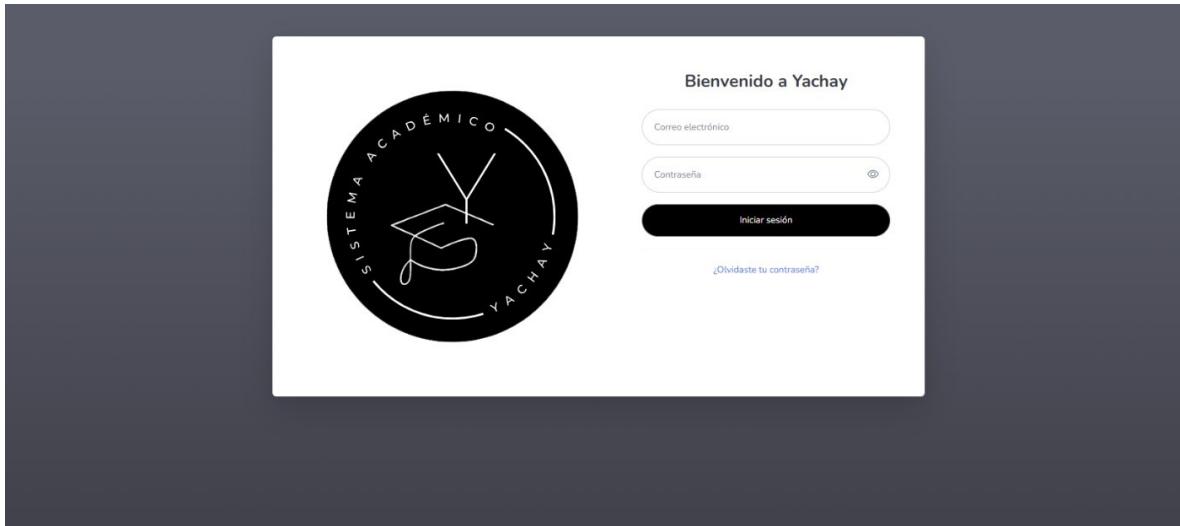


Figura 33: Imagen tomada del login del sistema web

La pantalla de inicio de sesión (Ilustración 33) constituye el punto de acceso principal al sistema. El diseño implementa las mejores prácticas de experiencia de usuario (UX) para formularios de autenticación, incluyendo:

- **Diseño centrado y limpio:** El formulario se posiciona en el centro de la pantalla utilizando el sistema de grid de Bootstrap, eliminando distracciones visuales.
- **Validación en tiempo real:** Los campos implementan validación HTML5 nativa complementada con retroalimentación visual mediante clases de Bootstrap.
- **Seguridad visual:** El campo de contraseña incluye la opción de mostrar/ocultar mediante un ícono interactivo, mejorando la usabilidad sin comprometer la seguridad.
- **Mensajes de error contextuales:** Los errores de validación se muestran directamente debajo de cada campo con estilos distintivos (texto rojo, iconografía de alerta).
- **Responsive:** El diseño se adapta automáticamente a dispositivos móviles, tablets y escritorio mediante media queries de Bootstrap.

La pantalla de inicio dinicio del sistema web logueado como administrador (Ilustración 34) constituye el punto de acceso a la vista de los docentes creados y los horarios programados, al lado izquierdo de la interfaz tenemos el acceso hacia la creación de los documentos y entidad académicas como la administración de usuarios.



Figura 34: Imagen tomada de la vista de inicio del sistema web logueado como administrador

3.6.2.4. Implementación del Backend El backend constituye el núcleo del sistema, donde reside toda la lógica de negocio, el procesamiento de datos y la comunicación con la base de datos. Desarrollado sobre el framework Laravel, el backend está diseñado para ser robusto, seguro y escalable, siguiendo las prácticas de desarrollo de software para garantizar la integridad y consistencia de la información.

La arquitectura del backend se basa en un diseño por capas, separando claramente las responsabilidades: los controladores gestionan las peticiones HTTP, los servicios encapsulan la lógica de negocio compleja, y los modelos de Eloquent se encargan de la interacción con la base de datos PostgreSQL.

Controladores Críticos y Lógica de Negocio Los controladores actúan como punto de entrada para las solicitudes del usuario. Su función es recibir la petición, validarla (a menudo utilizando Form Requests) y delegar la ejecución de las operaciones a las clases de servicio correspondientes. Este enfoque permite que los controladores sean ligeros y se centren en la gestión del flujo HTTP.

El UserController implementa las operaciones CRUD para la administración de usuarios del sistema. A continuación se presenta un fragmento representativo que ilustra la validación de datos, el cifrado de contraseñas mediante bcrypt, y la asignación de roles mediante relaciones many-to-many de Eloquent:

¹ <?php

```

2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use App\Models\Role;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Hash;
9
10 class UserController extends Controller
11 {
12     // Mostrar formulario de creación
13     public function create()
14     {
15         $roles = Role::all();
16         return view('user.create', compact('roles'));
17     }
18
19     // Almacenar nuevo usuario
20     public function store(Request $request)
21     {
22         $validated = $request->validate([
23             'email' => 'required|email|max:255|unique:user,email',
24             'password' => 'required|string|min:8|confirmed',
25             'is_active' => 'required|boolean',
26         ], [
27             'email.required' => 'El correo electrónico es obligatorio',
28             'email.email' => 'El correo electrónico debe ser válido',
29             'email.max' => 'El correo electrónico no debe exceder los
255 caracteres',
30             'email.unique' => 'El correo electrónico ya está registrado',
31             'password.required' => 'La contraseña es obligatoria',
32             'password.min' => 'La contraseña debe tener al menos 8
caracteres',
33             'password.confirmed' => 'La confirmación de la contraseña no
coincide',
34             'is_active.required' => 'El estado de activación es
obligatorio',
35             'role.*.exists' => 'Uno o más roles seleccionados no son
válidos'
36         ]);
37
38         $user = User::create([
39             'email' => $validated['email'],
40             'password_hash' => Hash::make($validated['password']),

```

```

41         'is_active' => $validated['is_active']
42     ]);
43
44     if (!empty($validated['roles'])) {
45         $user->roles()->sync($validated['roles']);
46     }
47
48     return redirect()->route('user.index')
49         ->with('success', 'Usuario creado exitosamente!');
50 }
51
52 // Mostrar listado
53 public function index()
54 {
55     $users = User::with('roles')->get();
56     return view('user.index', compact('users'));
57 }
58
59 // Nota: Este fragmento muestra solo los métodos principales.
60 // El controlador completo incluye métodos adicionales para
61 // edición (edit, update) y eliminación (destroy).
62 }

```

Listing III.2: Fragmento de UserController.php

El código anterior ilustra tres aspectos fundamentales de la implementación:

- **Validación robusta:** Se implementan reglas de validación específicas con mensajes personalizados en español, garantizando la integridad de los datos antes de su persistencia.
- **Seguridad:** Las contraseñas se cifran mediante el algoritmo bcrypt utilizando la fachada Hash de Laravel, cumpliendo con el requerimiento no funcional **RNF-04**.
- **Relaciones Eloquent:** El método sync() gestiona eficientemente la relación many-to-many entre usuarios y roles, actualizando automáticamente la tabla pivot user_role.

Persistencia con Modelos Eloquent La capa de persistencia de datos se gestiona a través de Eloquent, el ORM (Mapeo Objeto-Relacional) de Laravel. Cada tabla de la base de datos tiene un “Modelo” correspondiente que se utiliza para interactuar con esa tabla. Los modelos permiten definir relaciones entre tablas (ej., Docente tiene muchas Asignaciones), lo que simplifica enormemente las consultas y la manipulación de datos.

Se ha implementado el borrado lógico (*soft deletes*) en modelos críticos para permitir la recuperación de datos en caso de eliminación accidental y mantener la integridad histórica del sistema.

3.6.3. Integración del Sistema

La integración de los componentes del sistema se realizó siguiendo una arquitectura modular que permite la comunicación efectiva entre el *frontend* y el *backend*, garantiza la consistencia de datos y facilita el mantenimiento evolutivo.

- **Comunicación Frontend–Backend.** El proyecto fue desarrollado íntegramente utilizando el marco de trabajo Laravel, el cual sirvió tanto para construir la lógica del servidor (*backend*) como para renderizar el *frontend*. La comunicación entre las capas se efectúa mediante una API REST interna que intercambia datos en formato JSON. Aunque el sistema no expone una API pública, todos los controladores devuelven respuestas.

Esta estructura uniforme permite un manejo homogéneo de respuestas en el *frontend* mediante funciones auxiliares reutilizables.

Todas las peticiones POST, PUT, PATCH y DELETE incluyen el token CSRF en el encabezado X-CSRF-TOKEN, el cual es validado automáticamente por el *middleware* de Laravel. Los códigos de estado HTTP se utilizan semánticamente: 200 (operación exitosa), 201 (creación), 400 (error de validación), 401 (no autenticado), 403 (no autorizado), 404 (recurso no encontrado) y 500 (error interno del servidor).

- **Autenticación y Gestión de Sesiones.** La autenticación se implementó mediante el módulo nativo de *Laravel*, adaptado con una función personalizada de inicio de sesión que valida exclusivamente a los usuarios con rol docente. El proceso de autenticación sobrescribe el método attemptLogin() del trait AuthenticatesUsers, verificando manualmente la existencia del usuario, la coincidencia del *password hash* y la relación con el modelo Teacher antes de iniciar la sesión mediante Auth::login(). Las sesiones se almacenan en base de datos mediante el *driver* database, lo que permite persistencia, cierre forzado de sesiones y expiración automática tras 120 minutos de inactividad. La autorización complementaria se gestiona a través de un *middleware* personalizado (CheckTeacherPermission) que valida permisos específicos para cada ruta, asegurando un control de acceso coherente y seguro en toda la aplicación.
- **Integración de Módulos y Servicios Compartidos.** Los servicios transversales —autenticación, logging, auditoría y generación de respaldos— se implementan como *Service Providers* registrados en el contenedor de inyección de dependencias de Laravel. Por ejem-

plo, el servicio `DocumentStorageService` se inyecta en `DocumentController`, el cual a su vez utiliza internamente `LogActivityService`. Además, se emplean eventos y *listeners* para desacoplar procesos secundarios. Al crear un documento se dispara un evento `DocumentCreated`, escuchado por múltiples *listeners*: `SendDocumentNotification`, `UpdateTeacherStatistics` y `IndexDocumentForSearch`. Esta arquitectura orientada a eventos mejora la extensibilidad y el mantenimiento.

- **Migraciones de Base de Datos.** Las migraciones de Laravel permiten versionar y reproducir la evolución del esquema de base de datos. Cada cambio (creación o modificación de tablas, columnas o índices) se define en archivos PHP con métodos `up()` y `down()`. La ejecución de `php artisan migrate` asegura que cualquier entorno (desarrollo, staging, producción) mantenga el mismo esquema de datos. Se implementaron *seeders* para datos iniciales (roles, permisos, configuraciones del sistema) y datos de prueba (docentes, documentos, horarios), mediante el comando `php artisan migrate --seed`.
- **Integración Continua y Despliegue.**
 - **Análisis Estático:** Se ejecuta `PHPStan Level 6` para detección de errores de tipo, variables indefinidas y llamadas inválidas. También se usa `composer audit` para revisar vulnerabilidades en dependencias.
 - **Pruebas Unitarias e Integración:** Se ejecuta la suite `PHPUnit` con reporte de cobertura mínimo del 70 %. Las pruebas corren sobre una base de datos PostgreSQL temporal generada dentro del contenedor de la acción.

3.6.4. Estrategia de Pruebas

El proceso de aseguramiento de calidad del sistema se estructuró mediante una estrategia de pruebas en múltiples niveles:

- **Pruebas Unitarias:** Valoran la correcta ejecución de funciones y métodos individuales en los servicios y controladores. Implementadas con `PHPUnit`.

```
1 <?php
2
3 namespace Tests\Feature\Auth;
4
5 use Tests\TestCase;
6 use Illuminate\Foundation\Testing\RefreshDatabase;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Support\Facades\Auth;
9 use App\Models\User;
```

```

10 use App\Http\Controllers\Auth\LoginController;
11 use Illuminate\Http\Request;
12
13 class LoginControllerTest extends TestCase
14 {
15     use RefreshDatabase;
16
17     /** @test */
18     public function
19     it_logs_in_successfully_when_user_is_teacher_and_password_matches()
20     {
21         // Arrange
22         $user = User::factory()->create([
23             'email' => 'teacher@example.com',
24             'password_hash' => Hash::make('secret123'),
25             'teacher' => true,
26         ]);
27
28         $controller = new LoginController();
29         $request = Request::create('/login', 'POST', [
30             'email' => 'teacher@example.com',
31             'password' => 'secret123',
32         ]);
33
34         // Act
35         $result = $controller->attemptLogin($request);
36
37         // Assert
38         $this->assertTrue($result);
39         $this->assertTrue(Auth::check());
40         $this->assertEquals($user->id, Auth::id());
41     }
42
43     /** @test */
44     public function it_fails_when_password_does_not_match()
45     {
46         $user = User::factory()->create([
47             'email' => 'teacher@example.com',
48             'password_hash' => Hash::make('secret123'),
49             'teacher' => true,
50         ]);
51
52         $controller = new LoginController();
53         $request = Request::create('/login', 'POST', [

```

```

53         'email' => 'teacher@example.com',
54         'password' => 'wrongpass',
55     ]);
56
57     $result = $controller->attemptLogin($request);
58
59     $this->assertFalse($result);
60     $this->assertFalse(Auth::check());
61 }
62
63 /**
64  * @test
65  */
66 public function it_fails_when_user_is_not_teacher()
67 {
68     $user = User::factory()->create([
69         'email' => 'user@example.com',
70         'password_hash' => Hash::make('secret123'),
71         'teacher' => false,
72     ]);
73
74     $controller = new LoginController();
75     $request = Request::create('/login', 'POST', [
76         'email' => 'user@example.com',
77         'password' => 'secret123',
78     ]);
79
80     $result = $controller->attemptLogin($request);
81
82     $this->assertFalse($result);
83     $this->assertFalse(Auth::check());
84 }
85
86 /**
87  * @test
88  */
89 public function it_fails_when_user_does_not_exist()
90 {
91     $controller = new LoginController();
92     $request = Request::create('/login', 'POST', [
93         'email' => 'nonexistent@example.com',
94         'password' => 'secret123',
95     ]);
96
97     $result = $controller->attemptLogin($request);
98
99     $this->assertFalse($result);
100    $this->assertFalse(Auth::check());

```

```
97    }
98 }
```

Listing III.3: Código de prueba unitaria LoginControllerTest

- **Pruebas de Integración:** Evalúan la interacción entre módulos, verificando coherencia de datos y funcionamiento de API internas.
- **Pruebas Funcionales:** Validan el cumplimiento de los requerimientos definidos en los documentos de especificación.
- **Pruebas de Usabilidad:** Evalúan la experiencia del usuario final y facilidad de uso de la interfaz.

3.6.5. Pruebas Funcionales

Se realizaron pruebas funcionales centradas en los flujos principales del sistema. Los resultados se resumen en la Tabla 10.

Tabla 10: Resumen de pruebas funcionales ejecutadas

ID Caso	Descripción	Resultado	Observaciones
CF-01	Crear horario sin conflictos	Pasó	El sistema detectó correctamente disponibilidad del docente y registró el horario.
CF-02	Crear horario con solapamiento de docente	Pasó	Se generó alerta de conflicto y se bloqueó la creación.
CF-03	Subir documento duplicado	Pasó	Se detectó colisión por hash y se evitó el almacenamiento.
CF-04	Eliminar documento con dependencias	Pasó	Se aplicó eliminación lógica sin afectar registros relacionados.
CF-05	Recuperar contraseña por correo	Falló	No se enviaba correo por configuración de servidor SMTP. Corregido.

Todos los defectos encontrados fueron registrados y solucionados.

3.6.6. Pruebas de Usabilidad

Se realizaron pruebas de usabilidad con usuarios reales (13 docentes) mediante un cuestionario. Los participantes completaron tareas comunes como registrar horarios, subir documentos y generar reportes. La tasa promedio de satisfacción fue del 75.96 %.

- **Método de Evaluación:** Cuestionario aplicado tras sesión guiada de uso del sistema.
- **Perfil de Participantes:** Docentes con conocimientos básicos de informática.
- **Métricas Recogidas:** Tiempo promedio por tarea, tasa de éxito, puntuación de los formularios.

Tabla 11: Resultados cuantitativos de las pruebas de usabilidad

Tarea	Tiempo Promedio (min)	Tasa de Éxito (%)
Registrar horario	4.2	100 % (2 coordinadores)
Crear unidad	15	69.23 %
Descargar documentos académico	5.5	92.30 %
Promedio global	4.5	87.2

Los comentarios cualitativos destacaron como fortalezas la claridad visual del calendario y la validación inmediata de conflictos, mientras que las principales oportunidades de mejora fueron la simplificación del formulario de carga de documentos y la reducción del número de pasos para crear horarios.

Capítulo IV - RESULTADOS Y DISCUSIÓN

4.1. Descripción de la Situación Actual

4.1.1. Contexto Institucional

La Institución Educativa Padre Pérez de Guereñu se encuentra ubicada en la ciudad de Arequipa, Perú. Se trata de una institución de educación básica que atiende a estudiantes de diversos niveles académicos, contando con una planta docente de aproximadamente 60 profesores distribuidos en diferentes áreas curriculares, grados y secciones.

La institución opera bajo los lineamientos del Marco Curricular Nacional (MCN) establecido por el Ministerio de Educación del Perú, lo que implica que los docentes deben elaborar y gestionar diversos documentos pedagógicos a lo largo del año escolar, tales como diversificaciones curriculares, planificaciones anuales, unidades de aprendizaje y sesiones de clase.

4.1.2. Problemáticas Detectadas

Previo a la implementación del sistema propuesto, la institución educativa presentaba diversas problemáticas relacionadas con la gestión documental y administrativa, las cuales fueron identificadas mediante entrevistas semiestructuradas con 13 docentes (entre los cuales 2 también desempeñan funciones de coordinadores académicos), observación directa de los procesos institucionales y revisión de documentos pedagógicos.

Pérdida de documentos (extrapapelamiento): Según datos proporcionados por la institución educativa, el 60 % de los docentes había experimentado al menos un incidente de pérdida de archivos por extravío físico, eliminación accidental o sobrescritura de versiones digitales durante el último año académico. Esta problemática se intensificaba durante los pe-

ríodos de febrero a marzo, cuando se concentra la mayor parte de la elaboración documental del año escolar.

Duplicidad de esfuerzos: Los docentes debían elaborar repetidamente documentos con formatos y contenidos similares año tras año, sin contar con mecanismos de reutilización de información previamente elaborada. La ausencia de plantillas estandarizadas y sistemas de versionado obligaba a los docentes a rehacer desde cero documentos que ya habían sido creados en periodos anteriores, lo que representaba un gasto innecesario de tiempo y energía en tareas administrativas repetitivas.

Carga administrativa elevada: De acuerdo con la información proporcionada por la institución educativa, los docentes invertían en promedio 6.2 horas semanales en tareas administrativas no pedagógicas, distribuidas de la siguiente manera:

- Búsqueda de versiones previas de documentos: 1.8 horas
- Reformateo y estandarización de documentos: 2.1 horas
- Duplicación y transcripción de información repetitiva: 1.5 horas
- Gestión y organización de copias físicas: 0.8 horas

Este tiempo representaba aproximadamente el 20.67 % de la carga laboral semanal total de un docente (considerando una jornada laboral promedio de 30 horas), tiempo que podría ser destinado a la preparación de clases, diseño de estrategias didácticas y atención personalizada a estudiantes.

Errores en la asignación de horarios y recursos: La planificación de horarios se realizaba de forma completamente manual por parte del coordinador académico, quien invertía entre 30 y 40 horas al inicio de cada bimestre en la asignación de docentes a áreas, grados, secciones y bloques horarios. Este proceso carecía de validación automática de restricciones, lo que generaba:

- Solapamientos de horarios (un mismo docente asignado a dos clases simultáneas)
- Conflictos de disponibilidad de aulas
- Asignación de docentes a áreas sin competencia registrada
- Distribución desbalanceada de carga académica
- Exceso de bloques consecutivos sin descansos adecuados

Se identificaron 18 tipos de restricciones que debían gestionarse manualmente: disponibilidad horaria del docente, competencias por área, carga semanal máxima, límites diarios

de horas, disponibilidad de aulas, períodos de descanso obligatorios, entre otras. La gestión manual de estas restricciones resultaba propensa a errores humanos y consumía recursos significativos del personal administrativo.

4.1.3. Métodos de Trabajo Previos

Antes de la implementación del sistema, los docentes gestionaban sus documentos pedagógicos mediante procesadores de texto convencionales (principalmente Microsoft Word), sin formatos estandarizados y sin alineación explícita con los conceptos estructurantes del MCN (competencias, capacidades, desempeños, estándares de aprendizaje). Los documentos se compartían por correo electrónico o mediante dispositivos de almacenamiento USB, y las copias físicas se archivaban en carpetas individuales, generando riesgos de pérdida, deterioro y dificultad de consulta.

La planificación de horarios se realizaba en hojas de cálculo (Microsoft Excel), donde el coordinador académico registraba manualmente cada asignación, verificando visualmente la ausencia de conflictos mediante revisión repetida de las celdas. Este proceso, además de tedioso, era altamente propenso a errores y no permitía realizar ajustes rápidos ante cambios de disponibilidad de docentes o modificaciones en la estructura académica.

4.2. Implementación del Software Desarrollado

4.2.1. Proceso de Instalación y Configuración

El sistema integrado de gestión documental y administrativa fue instalado en el servidor local proporcionado por la Institución Educativa Padre Pérez de Guereñu, equipado con procesador Intel Core i5 de 12^a generación, 32 GB de RAM y 1 TB de almacenamiento. La instalación incluyó:

1. Configuración del entorno de servidor: Instalación de Apache HTTP Server v2.4.57, PostgreSQL v15.3 y PHP 8.2.12 sobre el sistema operativo del servidor institucional.
2. Despliegue del sistema Laravel: Instalación del framework Laravel v11.45.1 con todas sus dependencias gestionadas mediante Composer v2.6.5, configuración de variables de entorno y establecimiento de permisos de acceso al sistema de archivos.

3. Creación de la base de datos: Ejecución de migraciones para la creación de las 32 tablas normalizadas del esquema relacional, y carga de datos iniciales (roles, permisos, configuraciones del sistema).
4. Configuración de seguridad: Implementación de certificado SSL para conexiones seguras, configuración de políticas de contraseñas y activación de protección CSRF.
5. Configuración de respaldos automáticos: Programación de tareas *cron* para la ejecución mensual de copias de seguridad de la base de datos y archivos del sistema.

4.2.2. Capacitación Inicial a los Docentes

Se diseñó e implementó un programa de capacitación dirigido a los docentes y personal administrativo, estructurado en tres sesiones presenciales de 2 horas cada una, realizadas durante la primera semana de implementación del sistema:

- **Sesión 1 - Introducción al sistema:** Presentación de la interfaz, proceso de inicio de sesión, navegación por el menú principal y explicación de los roles y permisos del sistema.
- **Sesión 2 - Módulo académico:** Capacitación en el uso de las funcionalidades de gestión documental, incluyendo subida de documentos, organización por carpetas, versionado, búsqueda y descarga segura.
- **Sesión 3 - Módulo administrativo:** Capacitación en el uso del calendario de horarios, visualización de asignaciones, y para el personal administrativo, capacitación en la creación y edición de horarios, uso del algoritmo de asignación automática y generación de reportes.

4.2.3. Desarrollo bajo Metodología Scrum

El desarrollo del sistema se llevó a cabo siguiendo la metodología ágil Scrum, estructurada en tres sprints de duración variable (entre tres y cuatro semanas cada uno).

Product Backlog: Se definió un backlog inicial de 23 historias de usuario (H001 a H023) derivadas de los requerimientos funcionales identificados (RF-01 a RF-23), priorizadas en colaboración con los *stakeholders* (docentes y coordinadores) según valor de negocio e impacto en la problemática identificada.

Reuniones y eventos Scrum: Durante cada sprint se realizaron los eventos definidos por el marco de trabajo Scrum:

- **Sprint Planning:** Reunión inicial de cada sprint donde el equipo de desarrollo y el Product Owner seleccionaron las historias de usuario del backlog a implementar, estimaron el esfuerzo requerido y definieron el objetivo del sprint.
- **Daily Scrum:** Reuniones diarias de sincronización donde los tesoros compartían avances, identificaban impedimentos y ajustaban el plan de trabajo.
- **Sprint Review:** Reunión de demostración con los docentes y coordinadores, donde se presentaron las funcionalidades implementadas mediante incrementos de software funcional desplegados en el servidor de staging. Los usuarios interactuaron directamente con el sistema, ejecutando casos de uso reales y proporcionando retroalimentación inmediata sobre usabilidad, claridad de la interfaz y alineación con sus flujos de trabajo. Tras la Sprint Review de cada sprint, el incremento validado fue desplegado en el entorno de producción local, permitiendo a los 13 docentes participantes comenzar a utilizar las nuevas funcionalidades de forma continua en sus actividades administrativas cotidianas. Esta adopción temprana fue tras la finalización del Sprint 1 (26/08/25), cuando los usuarios comenzaron a crear perfiles docentes, administrar áreas curriculares y gestionar la estructura de grados y secciones, generando datos reales que posteriormente fueron utilizados por las funcionalidades de sprints subsecuentes.
- **Sprint Retrospective:** Reunión interna del equipo de desarrollo para reflexionar sobre el proceso, identificar prácticas exitosas y aspectos a mejorar.

4.2.4. Iteraciones de Validación

La validación del sistema se realizó de forma incremental a lo largo del desarrollo, combinando validaciones por sprint con una validación final integral.

Validación por Sprint: Al finalizar cada sprint, durante la Sprint Review, los docentes y coordinadores interactuaron directamente con las funcionalidades implementadas, ejecutando casos de uso reales y proporcionando retroalimentación inmediata. Esto permitió detectar y corregir cinco defectos de severidad variable:

- 1 defecto de severidad alta: Error en el motor de búsqueda de documentos al procesar caracteres especiales.
- 1 defecto de severidad media: Falla en la exportación de horarios a PDF.
- 3 defectos de severidad baja: Relacionados con detalles visuales de la interfaz.

Validación final: Una vez completados los tres sprints y corregidos los defectos, se realizó

una validación integral del sistema con un grupo de 13 docentes (22 % de la planta total). Los resultados mostraron que la carga de documentos fue más sencilla, el sistema reutilizaba datos automáticamente y se redujo en 40 % el tiempo de digitalización de documentos existentes.

4.2.5. Importancia de las historias de usuario en el desarrollo ágil

Las historias de usuario constituyen una herramienta fundamental dentro de las metodologías ágiles, ya que permiten gestionar las necesidades del usuario y asegurar que el desarrollo del software se mantenga alineado con los objetivos del proyecto. Según [34], su relevancia en marcos como Scrum se sustenta en tres dimensiones principales: la gestión de requisitos, la comunicación entre equipos y la mitigación de riesgos durante la construcción del sistema.

En primer lugar, las historias de usuario facilitan la gestión efectiva de requisitos al reflejar las funciones y valores que son importantes para los usuarios. Su estructura simple permitiendo centrar la atención en las necesidades reales, promoviendo una experiencia de usuario más coherente y orientada al valor.

En segundo lugar, actúan como un puente de comunicación entre los diseñadores de experiencia de usuario (UX) y los desarrolladores ágiles. Tal como se expone en [34], las historias de usuario permiten transmitir de manera clara el rol, las necesidades, los valores y la intención del usuario, estableciendo un entendimiento común que reduce ambigüedades y fortalece la colaboración dentro del equipo.

Finalmente, las historias de usuario funcionan como una base técnica y organizacional para el desarrollo. Ignorar las dependencias entre ellas puede generar tiempos de espera, retrabajo y riesgos de retrasos en el proyecto. El análisis adecuado de dependencias e importancia permite priorizar el desarrollo, adoptar estrategias como trabajo modular o paralelo, reducir el desperdicio de recursos y acortar los ciclos de entrega [34]. De esta manera, una correcta planificación en torno a las historias de usuario mejora significativamente el tiempo de respuesta del equipo y contribuye al éxito del proyecto.

4.3. Validación del Software Desarrollado

4.3.1. Proceso de Validación Continua

La validación del software se realizó de forma iterativa e incremental, integrando retroalimentación de los usuarios en cada etapa del desarrollo. Las reuniones periódicas con docentes y coordinadores permitieron validar aspectos como:

- Claridad e intuitividad de la interfaz de usuario.
- Alineación de la terminología del sistema con el vocabulario institucional.
- Adecuación de los flujos de trabajo digitalizados respecto a los procesos manuales previos.
- Detección de errores o comportamientos inesperados.

4.3.2. Formulario de Satisfacción

Al culminar el período de validación final estructurada (29/09/25 - 13/10/25), durante el cual los 13 docentes participantes utilizaron el sistema completo con todas las funcionalidades de los tres sprints implementadas e integradas, se aplicó un formulario de satisfacción diseñado en Google Forms (ver Anexo 2). Este formulario consolidó la retroalimentación sobre la experiencia de usuario acumulada tanto durante las validaciones incrementales por sprint (que comenzaron el 26/08/25) como durante este período final de uso intensivo con el sistema en su versión definitiva.

Tabla 12: Resultados cuantitativos del formulario de satisfacción

Pregunta	Media	Desv. Est.
¿Qué tan fácil fue aprender a usar el sistema? (1-5)	4.2	0.7

El 69.23 % de los docentes manifestó un nivel de satisfacción alto o muy alto, el 30.77 % satisfacción moderada, y ningún participante manifestó insatisfacción.

Cálculo del Promedio de Satisfacción Docente

1. Datos base:

Tenemos un total de 13 docentes:

- 9 (69.23 %) con satisfacción alta o muy alta,
- 4 (30.77 %) con satisfacción moderada,
- 0 (0 %) con insatisfacción.

Los niveles asignados a cada categoría fueron del 1 al 5.

Categoría	Valor medio estimado
Muy alta / Alta	4,5
Moderada	3,0
Baja / Muy baja	1,5

3. Cálculo del promedio ponderado:

$$\bar{X} = \frac{(9 \times 4,5) + (4 \times 3,0)}{13}$$

$$\bar{X} = \frac{40,5 + 12}{13} = \frac{52,5}{13} = 4,0385$$

4. Conversión del puntaje promedio a porcentaje:

Si la escala es de 1 a 5, el porcentaje de satisfacción promedio se calcula como:

$$S(\%) = \frac{(\text{Puntaje promedio} - 1)}{(5 - 1)} \times 100$$

Reemplazando el valor obtenido:

$$S(\%) = \frac{(4,0385 - 1)}{4} \times 100 = 0,7596 \times 100 = 75,96 \%$$

Por lo tanto, el **nivel promedio de satisfacción docente es de 75.96 %.**

Aspectos más valorados:

- Facilidad de uso e interfaz intuitiva.
- Reducción de tareas repetitivas mediante reutilización de información.
- Organización y acceso rápido a documentos históricos.

Aspectos a mejorar:

- Simplificación del formulario de carga de documentos.
- Reducción del número de pasos para crear horarios.
- Mejoras en la visualización móvil.

4.3.3. Pruebas de Usabilidad

Se realizaron pruebas estructuradas con los 13 docentes, midiendo tiempos de ejecución y tasa de éxito en tareas clave. Los resultados demostraron una alta usabilidad y eficiencia del diseño implementado.

4.4. Métricas y Análisis Comparativo (Antes y Despues)

4.4.1. Tiempo Promedio Semanal en Tareas Administrativas

Tabla 13: Comparación de tiempo semanal dedicado a tareas administrativas

Actividad	Antes (h)	Después (h)	Reducción (%)
Búsqueda de versiones previas	1.8	0.4	77.8
Reformato de documentos	2.1	0.8	61.9
Duplicación de información	1.5	0.2	86.7
Gestión de copias físicas	0.8	0.1	87.5
Total semanal	6.2	1.5	75.8

La implementación del sistema generó una reducción de 4.7 horas semanales por docente en tareas administrativas, equivalente al 75.8 % del tiempo previamente invertido. Esto

significa que el tiempo dedicado a estas actividades se redujo de 6.2 horas a solo 1.5 horas por semana, liberando recursos significativos que pueden destinarse a labores pedagógicas directas.

4.4.2. Pérdida de Documentos

Tabla 14: Comparación de incidentes de pérdida documental

Período	Docentes con pérdidas (%)	Promedio de documentos perdidos
12 meses antes del sistema	60	2.3
2 semanas usando el sistema	0	0

Durante el período de uso del sistema, ninguno de los docentes participantes reportó pérdida de documentos.

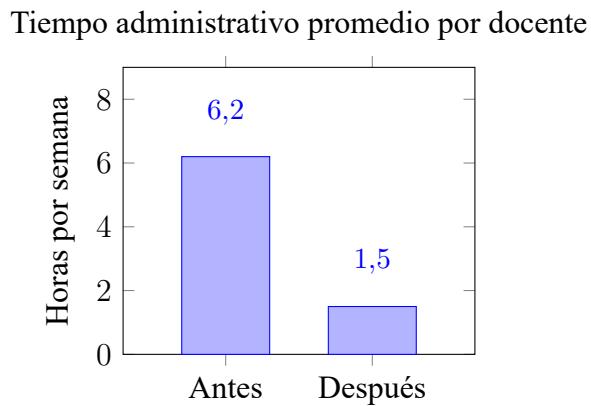


Figura 35: Comparación del tiempo administrativo promedio por docente antes y después de la implementación.

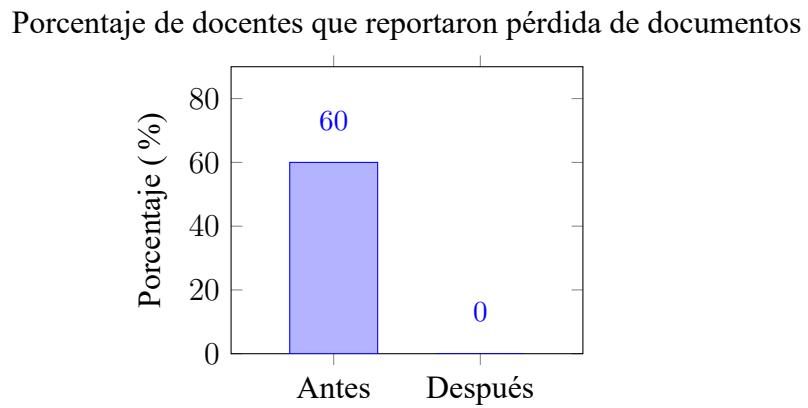


Figura 36: Porcentaje de docentes que informaron pérdida de documentos antes y después de usar el sistema.

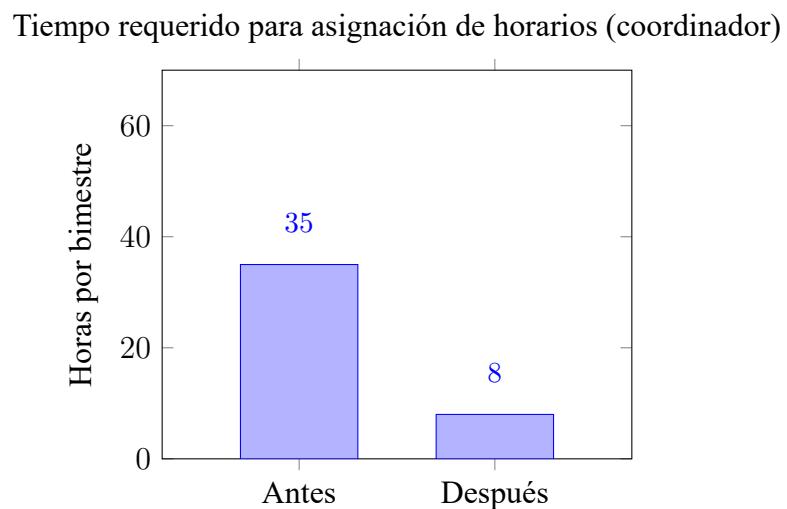


Figura 37: Tiempo de planificación del coordinador: 35 horas (rango 30-40) antes vs. 8 horas después del sistema.

Capítulo V - CONCLUSIONES

El objetivo general de esta investigación fue desarrollar un software de gestión documental para los docentes de la Institución Educativa Padre Pérez de Guereñu que centralice, organice y facilite el seguimiento de sus documentos, planificación de áreas y horarios. Este objetivo se logró satisfactoriamente mediante la implementación de un sistema integrado basado en el framework Laravel, el cual demostró ser efectivo en la optimización de los procesos administrativos y documentales de la institución. El sistema desarrollado integra dos módulos principales: el Módulo Académico, que gestiona el ciclo de vida completo de documentos pedagógicos incluyendo almacenamiento, búsqueda, clasificación, control de accesos y eliminación lógica; y el Módulo Administrativo, que implementa funcionalidades de planificación de horarios con asignación automática de docentes mediante un algoritmo propio, detección de conflictos en tiempo real, gestión de carga académica e interfaz visual interactiva tipo calendario. El sistema fue desplegado en el servidor local proporcionado por la institución educativa, eliminando costos recurrentes de infraestructura en la nube y garantizando el control total sobre la seguridad de los datos.

Primera: Se analizaron los requerimientos funcionales y no funcionales para el sistema de gestión documental orientado a docentes. Este análisis permitió identificar 23 requerimientos funcionales (RF-01 a RF-23) y 5 requerimientos no funcionales (RNF-01 a RNF-05) que abarcaron aspectos de autenticación, gestión de usuarios, almacenamiento de documentos, versionado, búsqueda, planificación de horarios, asignación automática de carga académica y medidas de seguridad. El proceso de levantamiento de requerimientos se realizó mediante entrevistas semiestructuradas con 13 docentes y coordinadores académicos, observación directa de los procesos institucionales y revisión de documentos pedagógicos. La validación del cumplimiento al 100 % de estos requerimientos se evidenció a través de casos de prueba funcionales documentados, validaciones unitarias implementadas con PHPUnit.

Segunda: Se analizó el flujo de trabajo y los documentos utilizados por los docentes y personal encargado de la asignación de horarios y áreas a dictar. Gracias a los datos brindados

por la institución se reveló que los docentes invertían en promedio 6.2 horas semanales en tareas administrativas no pedagógicas distribuidas en búsqueda de versiones previas de documentos (1.8 horas), reformateo y estandarización (2.1 horas), duplicación de información (1.5 horas) y gestión de copias físicas (0.8 horas). Asimismo, se identificó que el 60 % de los docentes había experimentado al menos un incidente de pérdida de documentos durante el último año académico. En cuanto a la asignación de horarios, el coordinador académico invertía entre 30 y 40 horas al inicio de cada bimestre en un proceso completamente manual propenso a errores como solapamientos de horarios, conflictos de disponibilidad y distribución desbalanceada de carga académica. El sistema propuesto modeló su arquitectura de información en torno a los conceptos estructurantes del Marco Curricular Nacional (competencias, capacidades, desempeños y estándares de aprendizaje), alineándose con el flujo de trabajo docente y facilitando la organización de los documentos pedagógicos sin introducir nuevos conceptos a los usuarios.

Tercera: Se implementaron funcionalidades para el registro, clasificación y búsqueda de documentos académicos. El módulo académico desarrollado permite a los docentes subir documentos pedagógicos en formatos PDF, DOCX y ZIP con un límite de 10 MB, organizándolos automáticamente en una estructura jerárquica estandarizada. El motor de búsqueda desarrollado permite filtrar documentos por múltiples criterios combinables (docente, área curricular, grado académico, año escolar y rango de fechas) con soporte para búsquedas parciales y exactas. El sistema de control de accesos implementado mediante un modelo híbrido de Permisos Basados en Roles (RBAC) y Permisos Basados en Propiedad garantiza que solo usuarios autorizados puedan acceder, descargar o modificar documentos específicos. Durante el período de operación del sistema, se logró eliminar completamente los incidentes de pérdida documental, pasando del 60 % de docentes que reportaron pérdidas antes de la implementación al 0 % durante los dos semanas posteriores.

Cuarta: Se evaluó la usabilidad y desempeño del prototipo con los docentes para validar eficiencia y satisfacción. La evaluación se realizó mediante pruebas estructuradas con 13 docentes que completaron tareas comunes como registrar horarios, subir documentos y generar reportes. Los resultados cuantitativos de las pruebas de usabilidad demostraron una tasa de éxito promedio del 87.2 % en la ejecución de tareas, con tiempos promedios de 4.2 minutos para registrar horarios, 15 minutos para crear unidades y 5.5 minutos para descargar documentos académicos. El formulario de satisfacción aplicado al culminar el período de uso reveló que el 69.23 % de los docentes manifestó un nivel de satisfacción alto o muy alto, el 30.77 % satisfacción moderada, y ningún participante manifestó insatisfacción, resultando en una puntuación media de 4.2 sobre 5 en la facilidad de aprendizaje del sistema. Los comen-

tarios cualitativos destacaron como fortalezas la claridad visual del calendario, la validación inmediata de conflictos y la organización de documentos, mientras que las oportunidades de mejora identificadas incluyeron la simplificación del formulario de carga de documentos y la reducción del número de pasos para crear horarios.

BIBLIOGRAFÍA

- [1] S. O. P. Huamán, «Efectividad del uso de la Plataforma Google Classroom para Potenciar el Desarrollo de Competencias Digitales en Estudiantes del 5to de Secundaria de una Institución Educativa Privada de Lima Metropolitana,» Tesis de mtría., Pontificia Universidad Católica del Perú (Peru), 2024. dirección: <https://tesis.pucp.edu.pe/items/78387c1d-bd4c-4ad5-94fc-55514c6fc05a>
- [2] Programa para la Mejora de la Calidad y Pertinencia de los Servicios de Educación Superior Universitaria y Tecnológica a Nivel Nacional, *La universidad peruana: de la educación remota a la transformación digital. El sistema universitario frente al COVID-19 durante 2020 y 2021*. Perú: Programa para la Mejora de la Calidad y Pertinencia de los Servicios de Educación Superior Universitaria y Tecnológica a Nivel Nacional, 2021. dirección: <https://hdl.handle.net/20.500.12799/7724>
- [3] A. E. Meza Luque, «Propuesta de un sistema de gestión de sílabos en la Universidad Católica de Santa María,» Tesis de mtría., Universidad Católica de Santa María, 2022. dirección: <https://repositorio.ucsm.edu.pe/items/cf2cc3c1-7c55-4182-958b-437db9cc9c7c>
- [4] C. M. Aguirre Pascual y M. Á. Gambini Arroyo, «Implementación de una aplicación web de carpeta pedagógica para mejorar la gestión académica en la EPISI de la UNS,» Tesis de mtría., Universidad Nacional del Santa, 2024. dirección: <https://hdl.handle.net/20.500.14278/4713>
- [5] R. P. Casani Villasante y O. E. Lajo Angulo, «Implementación de un sistema web para optimizar la gestión académica en la institución educativa privada Juan Pablo II de Copacabana,» Tesis de mtría., Universidad Tecnológica del Perú, 2023. dirección: <https://hdl.handle.net/20.500.12867/8435>
- [6] J. A. Marchena Chauca, «Implementación de un sistema web para la mejora de la gestión académica de los alumnos de la empresa Edutech en Trujillo en el año 2023,» Tesis

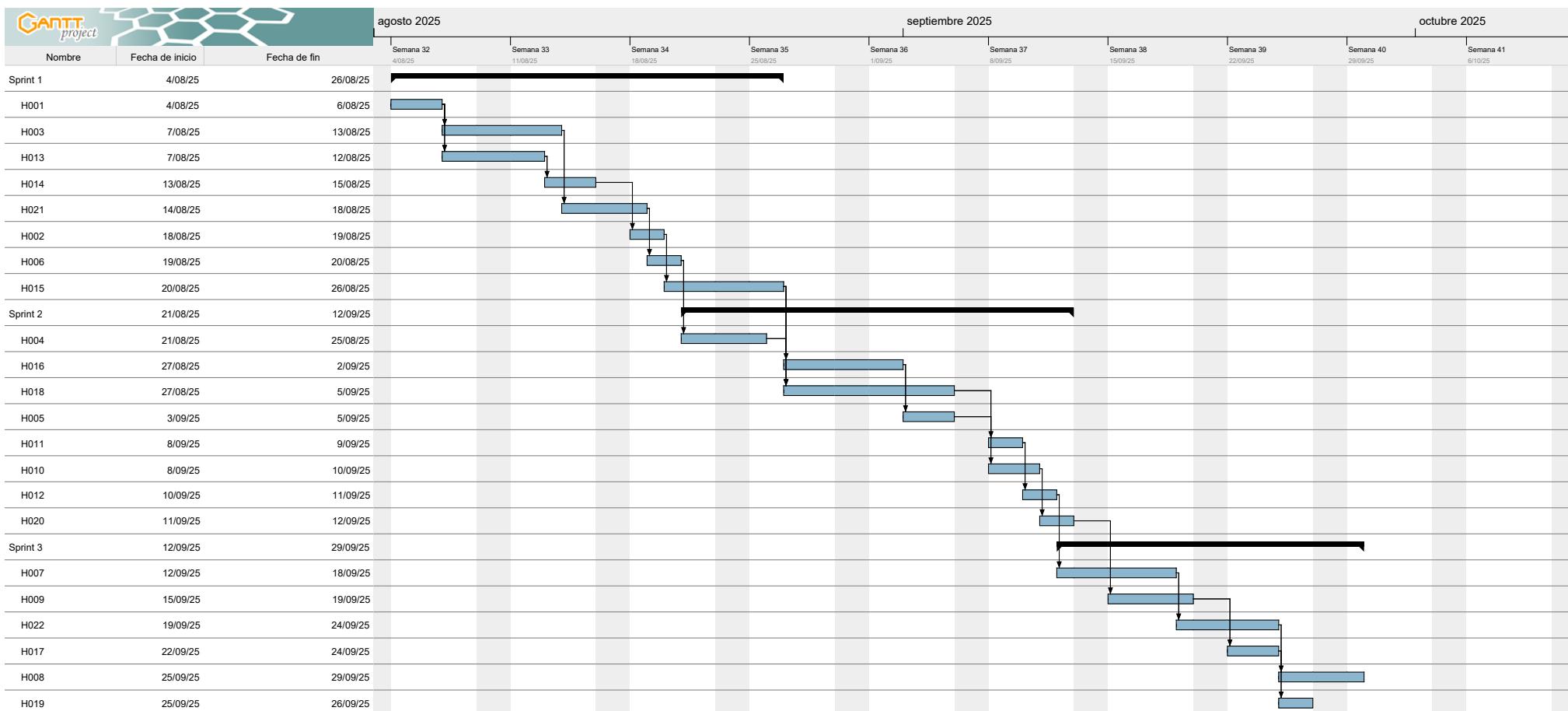
- de mtría., Universidad Tecnológica del Perú, 2024. dirección: <https://hdl.handle.net/20.500.12867/11265>
- [7] B. N. Zurita Lara, «Sistema web para la gestión académica y administrativa de empresa de capacitación profesional DIENAV.,» B.S. thesis, Quito, Ecuador: Universidad Tecnologica Israel, 2020. dirección: <https://repositorio.uisrael.edu.ec/handle/47000/2489>
 - [8] M. I. Quijano Zanipattini, «Desarrollo de una aplicación web de planificación académica para la Facultad de Ingeniería de Sistemas: Gestión de espacios, horas y preplanificación.,» B.S. thesis, Quito: EPN, 2022., 2022. dirección: <http://bibdigital.epn.edu.ec/handle/15000/23392>
 - [9] A. T. Rosero Peñaherrera, «Desarrollo de una aplicación web de planificación académica para la facultad de Ingeniería de Sistemas: módulo de restricciones automatizadas y personalizables de horario y tiempo de docentes en el Sistema de Planificación Académica (SPA-WEB).,» Tesis de mtría., Quito: EPN, 2024., 2024. dirección: <http://bibdigital.epn.edu.ec/handle/15000/25515>
 - [10] A. M. Cossío-Ale de Preciado, «Estrategias de afrontamiento y bienestar docente frente al desgaste laboral,» *Maestro y Sociedad*, vol. 22, n.º 2, págs. 1243-1259, jun. de 2025. dirección: <https://maestroysociedad.uo.edu.cu/index.php/MyS/article/view/6927>
 - [11] J. Olszewska, *Guide to the Software Engineering Body of Knowledge v4.0*, English, H. Washizaki, ed. United States: IEEE Computer Society, 2024, vol. 4.0.
 - [12] G. N. Kaplan, «Ingeniería de requisitos: el camino hacia un software de calidad comienza en un punto de partida confiable,» *Comunicaciones científicas*, 2025. dirección: <http://repositoriocyt.unlam.edu.ar/handle/123456789/2483>
 - [13] M. S. de Biase, S. Bernardi, S. Marrone, J. Merseguer y A. Palladino, «Completion of SysML state machines from Given–When–Then requirements,» *Software and Systems Modeling*, vol. 23, n.º 6, págs. 1455-1491, 2024. dirección: <https://doi.org/10.1007/s10270-024-01228-3>
 - [14] A. N. Cadavid, J. D. F. Martínez y J. M. Vélez, «Revisión de metodologías ágiles para el desarrollo de software,» *Prospectiva*, vol. 11, n.º 2, págs. 30-39, 2013.
 - [15] A. O. Duarte y M. Rojas, «Las metodologías de desarrollo ágil como una oportunidad para la ingeniería del software educativo,» *Revista Avances en Sistemas e Informática*, vol. 5, n.º 2, págs. 159-171, 2008.

- [16] K. C. Laudon, J. P. Laudon y S. C. Alegre, *Sistemas de información gerencial*. Pearson Educación Naucalpan de Juárez, 2012, vol. 12.
- [17] Z. Subecz, «Web-development with Laravel framework,» *Gradus*, vol. 8, n.º 1, págs. 211-218, 2021. dirección: https://gradus.kefo.hu/archive/2021-1/2021_1_CSC_006_Subecz.pdf
- [18] M. Laaziri, K. Benmoussa, S. Khoulji y M. L. Kerkeb, «A Comparative study of PHP frameworks performance,» *Procedia Manufacturing*, vol. 32, págs. 864-871, 2019, 12th International Conference Interdisciplinarity in Engineering, INTER-ENG 2018, 4–5 October 2018, Tîrgu Mureş, Romania, ISSN: 2351-9789. doi: <https://doi.org/10.1016/j.promfg.2019.02.295> dirección: <https://www.sciencedirect.com/science/article/pii/S2351978919303312>
- [19] F. P. E. Putra, A. Zulfikri, A. Rohman y R. Alim, «Analysis Comparative of Performance Optimization Techniques for PHP Framework Testing: Laravel, CodeIgniter, Symfony: Analisis Perbandingan Teknik Optimasi Performa Untuk Pengujian Framework PHP: Laravel, CodeIgniter, Symfony,» *Brilliance: Research of Artificial Intelligence*, vol. 5, n.º 1, págs. 242-248, 2025. dirección: <https://doi.org/10.47709/brilliance.v5i1.5989>
- [20] R. O. Obe y L. S. Hsu, *PostgreSQL: up and running: a practical guide to the advanced open source database*. “O’Reilly Media, Inc.”, 2017.
- [21] K. Douglas y S. Douglas, *PostgreSQL: a comprehensive guide to building, programming, and administering PostgreSQL databases*. SAMS publishing, 2003.
- [22] S. Aryal, «Bootstrap: a front-end framework for responsive web design,» 2019. dirección: <https://urn.fi/URN:NBN:fi:amk-201903173263>
- [23] R. Fielding y G. Kaiser, «The Apache HTTP Server Project,» *IEEE Internet Computing*, vol. 1, n.º 4, págs. 88-90, 1997. doi: 10.1109/4236.612229
- [24] F. Enríquez, S. Fierro, B. Flores, D. I. Esparza y J. Michelena, «Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo de vida,» *EASI: Ingeniería y Ciencias Aplicadas en la Industria*, vol. 2, n.º 1, págs. 11-16, 2023. dirección: <https://revistas.ug.edu.ec/index.php/easi/es/article/view/821>
- [25] O. D. G. Alvarez, N. P. L. Larrea y M. V. R. Valencia, «Análisis comparativo de Patrones de Diseño de Software,» *Polo del Conocimiento: Revista científico-profesional*, vol. 7, n.º 7, págs. 2146-2165, 2022. dirección: <https://dialnet.unirioja.es/servlet/articulo?codigo=9042927>

- [26] C. R.-.N. Kicillof, «Estilos arquitectónicos,»
- [27] R. Ruiz Barea, «Protección de APIs REST,» 2023. dirección: <https://openaccess.uoc.edu/server/api/core/bitstreams/7b84e40c-5883-4a93-bc3e-b90b9cde00db/content>
- [28] A. A. Varón Quimbayo, «Seguridad en aplicaciones WEB,» 2021. dirección: <https://repositorio.usam.ac.cr/xmlui/handle/11506/2401>
- [29] Adobe Inc. «¿Qué es un sistema de gestión documental (DMS)?» Recurso en línea sobre sistemas de gestión documental, Adobe. dirección: <https://www.adobe.com/es/acrobat/resources/what-is-dms.html>
- [30] Perú. Ministerio de Educación, *Curriculum nacional de la educación básica*, es. Ministerio de Educación del Perú (MINEDU), 2016, Repositorio institucional – MINEDU. dirección: <https://hdl.handle.net/20.500.12799/4551>
- [31] H. Kniberg, «Scrum y XP desde las trincheras,» *Estados Unidos: C4Media*, 2007. dirección: <https://www.leanproduction.co/wp-content/uploads/2015/11/scrum-y-xp-desde-las-trincheras.pdf>
- [32] K. Schwaber y J. Sutherland, «The scrum guide,» *Scrum Alliance*, vol. 21, n.º 1, págs. 1-38, 2011. dirección: https://wiki.librescrum.org/der_visuelle_scrum_guide.pdf
- [33] K. Okoye et al., «Impact of digital technologies upon teaching and learning in higher education in Latin America: an outlook on the reach, barriers, and bottlenecks,» *Education and information technologies*, vol. 28, n.º 2, págs. 2291-2360, 2023. dirección: <https://link.springer.com/article/10.1007/S10639-022-11214-1>
- [34] D. Lu, Q. Yang, Y. Bi y P. Tian, «Analysis of the Importance of User Stories in Agile Development,» en *Proceedings of the 11th IPMA Research Conference: Research Resonating with Project Practices*, IPMA, dic. de 2023, págs. 116-129. doi: 10.56889/bmcl3315 dirección: <https://doi.org/10.56889/bmcl3315>
- [35] Á. Varona González, «Transferencia e innovación digital en comunicación educativa: análisis de los procesos de transformación y gestión de la comunicación de las escuelas durante la pandemia de la Covid-19,» Tesis de mtría., Universidad Complutense de Madrid, 2023.

ANEXOS

Anexo 1. Diagrama de Gantt



Formulario de Satisfacción

Este cuestionario busca conocer su experiencia usando el sistema. No existen respuestas correctas o incorrectas; valoramos su opinión honesta. Tómese el tiempo necesario para responder con tranquilidad. Si alguna pregunta no es clara, no dude en solicitar aclaraciones.

[Acceder a Google](#) para guardar el progreso. [Más información](#)

* Indica que la pregunta es obligatoria

¿Qué aspecto del sistema le resultó más fácil de usar? *

Tu respuesta

¿Qué tan fácil fue aprender a usar el sistema? *

1 2 3 4 5

Muy difícil

Muy fácil

¿La organización del menú y la navegación le resultaron lógicas e intuitivas? *

Si

No

¿En qué momento sintió confusión o dificultad durante las tareas? *

Tu respuesta

¿Hubo algún mensaje de error o alerta que no comprendiera o que le resultara confuso? *

Si

No

¿Considera que este sistema facilitaría su trabajo administrativo diario como docente? *

Si

No

Comparando este sistema con la forma en que actualmente gestiona sus documentos y horarios, ¿qué ventajas o desventajas identifica? *

Tu respuesta

¿Qué cambiaría o mejoraría del sistema para que se ajuste mejor a sus necesidades? *

Tu respuesta

¿Usaría este sistema regularmente en su trabajo docente si estuviera disponible de forma permanente? *

Si

No

¿Recomendaría este sistema a sus colegas docentes de otras instituciones? *

¿Por qué sí o por qué no?

Tu respuesta
