



**UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ
EXTENSIÓN EN EL CARMEN
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**
Creada Ley No. 10 – Registro Oficial 313 de noviembre 13 de 1985

PROYECTO INTEGRADOR

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TECNOLOGÍAS DE LA INFORMACIÓN**

**APLICACIÓN WEB PARA LA PLANIFICACIÓN ACADÉMICA
DE HORARIOS EN LA CARRERA DE INGENIERÍA EN SOFTWARE
EN ULEAM EXTENSIÓN EL CARMEN.**

AUTOR/ES:


FUERES TIPANTUÑA WILMER HERNÁN

TUTOR:

ING. ROCÍO ALEXANDRA MENDOZA VILLAMAR, MG.

EL CARMEN, ENERO 2026

CERTIFICACIÓN DEL TUTOR

 Uleam EL OY ALFARO DE MANABÍ	NOMBRE DEL DOCUMENTO: CERTIFICADO DE TUTOR(A).	CÓDIGO: PAT-04-F-004
	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1 Página 1 de 1

CERTIFICACIÓN

En calidad de docente tutor(a) de la Extensión El Carmen de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

Haber dirigido, revisado y aprobado preliminarmente el Trabajo de Integración Curricular bajo la autoría del estudiante **Fueres Tipantuña Wilmer Hernan**, legalmente matriculado/a en la carrera de Tecnologías de la Información, período académico 2025(1)-2025(2), cumpliendo el total de 384 horas, cuyo tema del proyecto **Aplicación web para la planificación académica de horarios en la carrera de ingeniería en Software en Uleam extensión El Carmen.**

La presente investigación ha sido desarrollada en apego al cumplimiento de los requisitos académicos exigidos por el Reglamento de Régimen Académico y en concordancia con los lineamientos internos de la opción de titulación en mención, reuniendo y cumpliendo con los méritos académicos, científicos y formales, y la originalidad del mismo, requisitos suficientes para ser sometida a la evaluación del tribunal de titulación que designe la autoridad competente.

Particular que certifico para los fines consiguientes, salvo disposición de Ley en contrario.

Lugar, El Carmen 29 de enero del 2026.

Lo certifico,



Ing. Rocio Mendoza Villamar, Mtr.
Docente Tutora
Área: Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

TRIBUNAL DE SUSTENTACIÓN



Universidad Laica Eloy Alfaro de Manabí
Extensión El Carmen
Carrera de Ingeniería en Tecnologías de la Información

TRIBUNAL DE SUSTENTACIÓN

Título del Trabajo de Titulación:

Aplicación web para la planificación académica de horarios en la carrera de ingeniería en Software en ULEAM Extensión El Carmen.

Modalidad:

Proyector Integrador

Autor:

Fueres Tipantuña Wilmer Hernan

Tutor:

Ing. Mendoza Villamar Roció Alexandra, Mg.

Tribunal de Sustentación:

- **Presidente:** Ing. Reascos Pinchao Raul Saed, Mg.

- **Miembro:** Ing. Arévalo Hermida Rómulo Danilo, Mg

- **Miembro:** Ing. López Rodríguez Carlos Vinicio, Mg.

Fecha de Sustentación:

20 de febrero de 2026

**UNIVERSIDAD LAICA “ELOY ALFARO” DE MANABÍ
EXTENSIÓN EN EL CARMEN**



DECLARACIÓN DE AUTORÍA

La responsabilidad del contenido de este Trabajo de titulación, cuyo tema es: APLICACIÓN WEB PARA LA PLANIFICACIÓN ACADÉMICA DE HORARIOS EN LA CARRERA DE INGENIERÍA EN SOFTWARE EN ULEAM EXTENSIÓN EL CARMEN, corresponde exclusivamente a: FUERES TIPANTUÑA WILMER HERNAN con CI. 235046534-6, y los derechos patrimoniales de la misma corresponden a la Universidad Laica Eloy Alfaro de Manabí.

Fures Tipantuña Wilmer Hernán

C.I. 235046534-6

Dedicatoria

A Dios, por haber sido mi guía constante, por fortalecer mi fe en los momentos más difíciles y por regalarme la oportunidad de culminar esta etapa tan importante de mi vida.

A mis padres, por sus sacrificios, enseñanzas y por haber sembrado en mí los principios que hoy me sostienen. Su ejemplo y apoyo han sido fundamentales para alcanzar este logro.

A mi familia, especialmente a mi esposa Gabriela, compañera incondicional de vida, cuyo amor, paciencia y comprensión me han motivado en cada paso. A mis hijas, Valentina y Katherín, mi mayor inspiración y motor, porque este esfuerzo está pensado en ustedes, para dejarles una raíz firme de valores, esfuerzo y amor que guíe su crecimiento. Son mi vida entera.

Gracias a todos por ser parte de este camino. Este logro no es solo mío, es de cada uno de ustedes.

Wilmer Fures

Agradecimiento

Agradezco, en primer lugar, a Dios, por haberme dado la vida, la salud, la fortaleza y la sabiduría necesarias para enfrentar cada etapa de esta carrera. Sin su guía constante, este logro no habría sido posible.

A mi querida esposa Gabriela Cusme, por su amor, paciencia y apoyo incondicional durante cada jornada de esfuerzo. Gracias por creer en mí incluso cuando yo dudaba. A mis hijas, Valentina Fures y Katherin Fures, por ser mi motor diario y por inspirarme a seguir creciendo para ser un mejor ejemplo para ustedes.

A mis padres, por inculcarme desde pequeño los valores del esfuerzo, la honestidad y la perseverancia. A mis hermanos y amigos, quienes, con su palabra oportuna, compañía y aliento, fueron parte esencial de este proceso.

En especial, a la memoria de mi hermano Paco Fures, cuyo apoyo moral y cariño me siguen acompañando desde el cielo. Su confianza en mí ha sido un faro en los momentos más difíciles, y este logro también le pertenece.

Extiendo mi agradecimiento a los docentes de la carrera de Ingeniería en Software de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, en especial a mi tutora de tesis Ing. Rocío Mendoza, por su dedicación, compromiso y por compartir sus conocimientos con vocación. A todos mis compañeros de aula, gracias por los momentos compartidos, el trabajo en equipo y la amistad construida a lo largo del camino.

Finalmente, agradezco a todas aquellas personas que, de una u otra manera, contribuyeron a que este proyecto se concrete. Este logro es resultado de un camino compartido, lleno de aprendizaje, esfuerzo y esperanza.

El autor

ÍNDICE GENERAL

CERTIFICACIÓN DEL TUTOR.....	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN DE AUTORÍA	V
Dedicatoria	VI
Agradecimiento	VII
ÍNDICE GENERAL.....	VIII
Índice tablas.....	XVII
Índice gráficos e ilustraciones	XIX
INDICE DE ANEXOS	XXI
Resumen	XXII
Abstract.....	XXIII
CAPÍTULO I.....	1
INTRODUCCIÓN.....	1
1.1 Introducción	1
1.2 Presentación del tema	2
1.3 Ubicación y contextualización de la problemática.....	2
1.4 Planteamiento del problema	2
1.4.1 Problematización	2
1.4.2 Génesis del problema	3
1.4.3 Estado actual del problema.....	3
1.5 Diagrama causa – efecto del problema.....	4
1.6 Objetivos.....	5
1.6.1 Objetivo general	5
1.6.2 Objetivos específicos.....	5

1.6.3. Hipótesis	5
1.7 Justificación	5
1.8 Impactos esperados.....	6
1.8.1 Impacto tecnológico	6
1.8.2 Impacto social.....	7
1.8.3 Impacto ecológico	8
CAPÍTULO II.....	9
MARCO TEÓRICO	9
2.1 Antecedentes históricos	9
2.2 Antecedentes de investigaciones relacionadas al tema presentado	10
2.3 Definiciones conceptuales	10
2.3.1 Aplicación web.....	10
2.3.1.1 Diseño de una aplicación web.	11
2.3.1.2 Evolución de las aplicaciones web	11
2.3.1.3 Procesos para diseñar una aplicación web.....	12
2.3.1.4 Tecnologías asociadas al desarrollo web.....	13
2.3.1.5 Ventajas y desafíos	14
2.3.1.6 Planificación académica de horarios	14
2.3.1.7 Características del proceso de planificación.....	15
2.3.2 Problemas frecuentes en la planificación manual	15
2.3.2.1 Ventajas de la planificación automatizada	15
2.3.2.2 Perspectiva pedagógica de la planificación	16
2.3.2.3 Componentes clave en la planificación académica	16
2.3.2.4 Herramientas tecnológicas utilizadas	16

2.3.2.5 Impacto en la gestión académica.....	16
2.3.2.6 Planificación académica en las instituciones de educación superior.....	17
2.3.2.7 Definición especializada del proceso	17
2.3.2.8 Componentes clave.....	18
2.3.2.9 Diseño de implementación en sistemas automatizados.....	18
2.3.3 Implementación tecnológica y módulos	18
2.3.3.1 Seguridad en la implementación	19
2.3.3.2 Procesos de validación y ajustes en tiempo real.....	19
2.3.3.3 Indicadores de rendimiento del horario académico.....	19
2.3.3.4 Integración con otros sistemas institucionales	20
2.3.3.5 Escalabilidad y personalización del sistema.....	20
2.3.3.6 Formación del personal y adopción institucional.....	20
2.3.3.7 Simulación de escenarios académicos.....	20
2.3.3.8 Accesibilidad y experiencia del usuario (UX).....	21
2.3.3.9 Gestión de incidencias planificadas.....	21
2.3.4 Auditoría y trazabilidad.....	21
2.3.4.1 Gestión del capital humano docente.....	21
2.3.4.2 Evaluación de la eficiencia horaria.....	21
2.3.4.3 Automatización del proceso de planificación.....	21
2.3.4.4 Seguridad y control de acceso en la planificación.....	22
2.3.4.5 Integración con sistemas institucionales	22
2.3.4.6 La planificación académica como proceso estratégico para la sostenibilidad institucional.....	22
2.3.4.7 Planificación académica orientada al estudiante mediante TIC	22
2.3.5 Metodología propuesta	22
Modelo en Cascada.....	22

2.3.5.1 Definición y características del modelo en cascada	23
2.3.5.2 Fases del modelo en cascada aplicadas al proyecto	23
2.3.5.3 Justificación del modelo en cascada según el contexto del proyecto ...	24
2.4 Conclusiones relacionadas al marco teórico en referencia al tema planteado....	25
CAPÍTULO III	26
MARCO INVESTIGATIVO.....	26
3.1 Introducción.....	26
3.2 Tipos de investigación.....	26
3.2.1 Investigación bibliográfica	26
3.2.2 Investigación de campo	26
3.3 Enfoques y métodos de la investigación	27
3.3.1 Enfoque cualitativo.....	27
3.3.1.2 Enfoque cuantitativo.....	27
3.3.1.3 Método deductivo:.....	27
3.3.1.4 Método analítico:.....	27
3.4 Fuentes de información de datos	28
3.4.1 Fuentes primarias.....	28
3.4.2 Fuentes secundarias	28
3.5 Estrategia operacional para la recolección de datos	28
3.5.1 Población	28
3.5.2 Segmentación	29
3.5.3 Censo poblacional.	29
3.6 Análisis de las herramientas de recolección de datos a utilizar.....	29
3.6.1 Encuesta.....	30
3.6.2 Entrevista.....	30

3.6.3 Estructura del instrumento de recolección de datos aplicados	30
3.6.3.1 Cuestionario para docentes.....	30
3.6.3.2 Entrevista dirigida al coordinador de la carrera	30
3.6.4 Plan de recolección de datos.....	31
3.6.5 Análisis y presentación de resultados.....	31
3.6.5.1 Tabulación y análisis de los datos	31
3.6.5.2 Encuesta.....	32
3.6.5.3 Entrevista.....	35
3.6.5.4 Presentación y descripción de los resultados obtenidos	38
3.6.5.5 Informe final del análisis de los datos	38
CAPÍTULO IV	39
MARCO PROPOSITIVO	39
4.1 Introducción.....	39
4.2 Descripción de la propuesta.....	39
4.3 Determinación de recursos	40
4.3.1 Humanos.....	40
4.3.2 Tecnológicos.....	40
4.3.2.1 Recursos de hardware.....	40
4.3.2.2 Recursos de software	41
4.3.3 Económicos	41
4.4 Etapas de acción para el desarrollo de la propuesta	42
4.4.1 Fase I: Requisitos	42
4.4.1.1 Recopilación de Requisitos	42
4.4.1.2 Análisis de Documentos	43
4.4.1.3. Análisis de Requisitos	43

4.4.1.4. Priorización de Requisitos.....	43
4.4.1.5 Priorización de requisitos funcionales.....	44
4.4.1.6 Priorización de requisitos no funcionales.....	44
4.4.2. Fase II: Análisis.....	44
4.4.2.1. Análisis de Sistemas:.....	44
4.4.2.1.1 Casos de uso principales.....	45
4.4.2.1.3 Descripción gráfica.....	45
4.4.2.1.4 Casos de uso del sistema	45
Caso de uso 1: Iniciar sesión	45
Caso de uso 2: Registrar información académica.....	46
Caso de uso 3: Generar horario (automático o manual).....	46
Caso de uso 4: Publicar y consultar horario	46
4.4.2.1.5 Diagramas de secuencia	47
Diagrama 1: Login de Usuario	47
Diagrama 2: Crear Horario.....	48
Diagrama 4: Generar Reporte PDF	49
4.4.2.1.6 Diagramas de estado.....	50
Diagrama 1: Ciclo de Vida de un Horario.....	50
Diagrama 2: Ciclo de Vida de un Usuario.....	51
Diagrama 3: Estado de Asignación de Materia	52
Diagrama 4: Estado de Período Académico	53
Diagrama 6: Detección y Resolución de Conflictos.....	54

4.4.2.2. Especificación de Requisitos.....	55
4.4.2.2.1. Requisitos Funcionales.....	55
4.4.2.2.2. Requisitos No Funcionales.....	56
4.4.3. Fase III: Diseño.....	56
4.4.3.1. Diseño del Sistema:.....	56
4.4.3.2. Arquitectura del Sistema.....	57
4.4.3.3. Diseño de Bases de Datos.....	58
4.4.3.4. Diseño de la Interfaz de Usuario.....	58
4.4.3.5. Diseño de UX.....	58
a) Colores:.....	58
b) Tipografía:.....	59
4.4.3.6. Prototipos.....	60
Mapa de navegación:.....	60
4.4.3.7 Interfaces de entrada:.....	60
4.4.3.7.1 Pantalla de Inicio de Sesión.....	60
4.4.3.7.2 Dashboard del Coordinador.....	61
4.4.3.7.3 Gestión de Periodos Académicos.....	61
4.4.3.7.4 Gestión de Información Académica.....	62
4.4.3.7.5. Generación Automática de Horarios.....	62
4.4.3.7.6 Editor de Horarios (horario manual).....	63
4.4.3.7.7 Gestión de Horarios.....	63
4.4.3.7.8 Módulo de Reportes - Estadísticas Generales.....	64
4.4.3.7.9 Consulta Histórica.....	64

4.4.3.8 Dashboard del Docente.....	65
4.4.3.8.1 Panel del Docente - Mi Horario Personal.....	65
4.4.3.8.2 Sistema de Notificaciones	66
4.4.3.8.3 Iconografía del Sistema	66
1. PANTALLA DE LOGIN	66
2. DASHBOARD DEL DOCENTE.....	67
3. DASHBOARD DEL COORDINADOR.....	67
Menú Lateral Izquierdo:	67
4.4.3.9 Conclusiones sobre Iconografía	68
4.4.4. Fase IV: Implementación	68
4.4.4.1. Codificación:	68
4.4.4.2. Programación web en el frontend y backend:	68
4.4.4.3 Herramientas y versiones:	69
4.4.4.3 Revisión de Código	70
4.4.5. Fase V: Pruebas	77
4.4.5.1. Pruebas de Sistema.....	77
4.4.5.2 Pruebas Funcionales	78
4.4.5.3 Pruebas de Rendimiento	82
4.4.5.4 Pruebas de Aceptación:	85
4.4.5.5 Validación con el Cliente	85
4.4.6. Fase VI: Mantenimiento	93
4.4.6.1 Implementación	93
4.4.6.1.1 Sistema en el hosting	93
4.4.6.1.2. Capacitación de usuarios	94

1. Coordinador Académico: Administración Completa del Sistema.....	94
2. Docentes: Consulta y Gestión Personal.....	95
Metodología de Capacitación	95
CAPÍTULO V	96
EVALUACIÓN DE RESULTADOS.....	96
5.1 Introducción.....	96
5.2 Presentación y monitoreo de resultados	96
5.2.1 Proceso Manual de Creación de Horarios	96
5.2.4 Proceso Manual de Resolución de Conflictos	97
5.2.5 Sistema Digital de Resolución de Conflictos	98
5.3 Interpretación objetiva.....	98
CAPÍTULO VI	99
CONCLUSIONES Y RECOMENDACIONES	99
6.1 Conclusiones.....	99
6.2 Recomendaciones	100
Bibliografía.....	101
Anexos.....	106
Glosario	112

Índice tablas

Tabla 1: Respuestas de la encuesta	32
Tabla 2: Entrevista dirigida al coordinador de la carrera de software	35
Tabla 3: Recursos económicos	41
Tabla 4: Especificación de requisitos funcionales	55
Tabla 5: Especificación de requisitos no funcionales	56
Tabla 6: Íconos del login	66
Tabla 7: Íconos del dashboard de docentes	67
Tabla 8: Íconos de dashboard del coordinador	67
Tabla 9: Pruebas funcionales de login del sistema	78
Tabla 10: Pruebas funcionales de gestión de usuarios	78
Tabla 11: Pruebas funcionales de gestión de carreras	79
Tabla 12: Pruebas funcionales de gestión de niveles	79
Tabla 13. Pruebas funcionales de gestión de paralelos	80
Tabla 14: Pruebas funcionales de gestión de aulas	80
Tabla 15: Pruebas funcionales de gestión de materias	80
Tabla 16: Pruebas funcionales de gestión de horarios	81
Tabla 17: Pruebas funcionales de gestión de periodos	82
Tabla 18: Pruebas de rendimiento de optimización de rendimiento	82
Tabla 19: Pruebas de rendimiento tiempos de respuesta del sistema	83
Tabla 20: Pruebas de rendimiento comportamiento bajo carga concurrente	84
Tabla 21: Pruebas de rendimiento uso de recursos del sistema	85
Tabla 22: Validación con el cliente de login del sistema	85
Tabla 23. Validación con el cliente del dashboard del coordinador	86
Tabla 24: Validación con el cliente del dashboard del docente	87
Tabla 25. Validación con el cliente de formulario de gestión de usuarios	88
Tabla 26: Validación con el cliente de formulario de gestión de carreras	89
Tabla 27: Validación con el cliente formulario de gestión de niveles	90
Tabla 28: Validación con el cliente formulario de gestión de aulas	90
Tabla 29: Validación con el cliente formulario de gestión de docentes	91
Tabla 30: Validación con el cliente formulario de gestión de materias	92
Tabla 31: Presentación y monitoreo de creación de horarios manual	96
Tabla 32: Presentación y monitoreo de creación de horarios en el sistema	97

Tabla 33: Presentación y monitoreo de resolución de conflictos manual.....	97
Tabla 34: Presentación y monitoreo de resolución de conflictos en el sistema	98

Índice gráficos e ilustraciones

Ilustración 1: Diagrama de causa-efecto	4
Ilustración 2: Sistema de planificación de horarios	45
Ilustración 3: Caso de uso iniciar sesión	45
Ilustración 4: Caso de uso registrar información académica	46
Ilustración 5: Caso de uso generar horario automático o manual	46
Ilustración 6: Caso de uso publicar y consultar horario	46
Ilustración 7: Diagrama de secuencia login de usuario	47
Ilustración 8: Diagrama de secuencia de crear horario	48
Ilustración 9: Diagrama de secuencia generar reporte	49
Ilustración 10: Diagrama de estado de ciclo de vida de un horario	50
Ilustración 11: Diagrama de estado de ciclo de vida de un usuario	51
Ilustración 12: Diagrama de estado de asignación de materia a horario	52
Ilustración 13: Diagrama de estado de estado de periodo académico	53
Ilustración 14: Diagrama de estado de detección y resolución de conflictos	54
Ilustración 15: Arquitectura del sistema modelo de 3 capas	57
Ilustración 16: Diseño de base de datos Modelo entidad relación	58
Ilustración 17: Mapa de navegación del sistema	60
Ilustración 18: Interfaz de inicio de sesión	60
Ilustración 19: Interfaz dashboard del coordinador	61
Ilustración 20: Interfaz de periodos académicos	61
Ilustración 21: Interfaz de información académica de docentes	62
Ilustración 22: Interfaz de generador automático de horarios	62
Ilustración 23: Interfaz para editar horarios	63
Ilustración 24: Interfaz de gestión de horarios	63
Ilustración 25: Interfaz para generar reportes estadísticos	64
Ilustración 26: Interfaz de consulta histórica de horarios	64
Ilustración 27: Interfaz del dashboard de docentes	65
Ilustración 28: Interfaz de horario personal del docente	65
Ilustración 29: Interfaz de notificaciones	66
Ilustración 30: Código de conexión a la base de datos	70
Ilustración 31: Código de verificar credenciales de inicio de sesión	70
Ilustración 32: Código de detección de conflictos	71

Ilustración 33: Código de generación de reportes pdf	72
Ilustración 34: Código para insertar niveles	72
Ilustración 35: Código de activación de periodos	73
Ilustración 36: Código para filtrar materias.....	73
Ilustración 37: Código para filtrar materias.....	73
Ilustración 38: Código para crear docente.....	74
Ilustración 39: Código para crear nueva materia	74
Ilustración 40: Código para verificar tokens jwt	75
Ilustración 41: Código para restringir roles	75
Ilustración 42: Código para crear nueva aula	76
Ilustración 43: Código para eliminar todos los horarios	76
Ilustración 44: Código para activar o desactivar usuarios.....	77
Ilustración 45: Hosting para alojamiento de aplicaciones web	93
Ilustración 46: Sección de file manager del hosting	94

INDICE DE ANEXOS

Anexo A. Aprobación de tema.....	106
Anexo B. Instrumento entrevista	107
Anexo C. Instrumento encuesta	108
Anexo D. Entrevista al coordinador	110
Anexo E. Certificado de coincidencia académica (emitido por tutor del sistema antiplagio.....	111

Resumen

La presente investigación tiene como objetivo el desarrollo de una aplicación web para la planificación académica de horarios en la carrera de Ingeniería en Software de la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, con la finalidad de optimizar los procesos administrativos académicos que actualmente se realizan de forma manual. El estudio parte del análisis de la problemática existente, caracterizada por el alto consumo de tiempo, errores humanos frecuentes, conflictos de horarios entre docentes y aulas, y la falta de un sistema centralizado para la consulta y gestión de horarios, apoyándose en una investigación bibliográfica para sustentar teóricamente el proyecto y en una investigación de campo mediante la recolección de información directa de los actores involucrados. Para dar solución a esta situación, se aplicó una metodología de desarrollo de software basada en el modelo en cascada, adecuada al carácter individual del proyecto y a la definición previa de los requisitos. La aplicación web fue desarrollada bajo una arquitectura de tres capas, utilizando tecnologías web modernas para la gestión de usuarios, carreras, niveles, paralelos, jornadas, materias, docentes, aulas y períodos académicos, permitiendo la generación automática y manual de horarios con validaciones en tiempo real. Asimismo, el sistema incorpora mecanismos de seguridad, control de acceso por roles y generación de reportes académicos. Los resultados obtenidos evidencian una mejora significativa en la eficiencia de los procesos, logrando una reducción promedio del 94,15 % en los tiempos de ejecución, además de una notable disminución de errores y una mejora en la experiencia de los usuarios. Finalmente, se concluye que la implementación del sistema contribuye a la modernización de la gestión académica y representa una solución tecnológica viable y escalable para la institución.

Palabras claves: Planificación académica, Aplicación web, Gestión de horarios, Automatización, Educación superior, Sistemas de información.

Abstract

This research aims to develop a web application for academic scheduling in the Software Engineering program at the Eloy Alfaro Lay University of Manabí, El Carmen Extension, in order to optimize the academic administrative processes that are currently performed manually. The study begins with an analysis of the existing problems, characterized by high time consumption, frequent human errors, scheduling conflicts between professors and classrooms, and the lack of a centralized system for consulting and managing schedules. It is supported by bibliographic research to provide a theoretical framework for the project and by field research through the collection of direct information from the stakeholders involved. To address this situation, a software development methodology based on the waterfall model was applied, adapted to the specific nature of the project and the previously defined requirements. The web application was developed using a three-tier architecture, employing modern web technologies for managing users, programs, levels, sections, shifts, subjects, professors, classrooms, and academic periods, allowing for the automatic and manual generation of schedules with real-time validation. The system also incorporates security mechanisms, role-based access control, and the generation of academic reports. The results obtained demonstrate a significant improvement in process efficiency, achieving an average reduction of 94.15% in execution times, as well as a notable decrease in errors and an improved user experience. In conclusion, the implementation of the system contributes to the modernization of academic management and represents a viable and scalable technological solution for the institution.

Keywords: Academic planning, Web application, Schedule management, Automation, Higher education, Information systems.

CAPÍTULO I

INTRODUCCIÓN

1.1 Introducción

La elaboración de horarios académicos de la universidad es uno de los procesos administrativos más estrictos para las instituciones de educación superior debido a su importante papel en la organización académica, la gestión de recursos y en la garantía de calidad del proceso educativo y la falta de planificación puede resultar en conflictos de horarios, sobrecarga excesiva de los docentes, utilización ineficaz de las aulas y retraso en el inicio de las actividades académicas, lo que puede afectar tanto a educadores como a estudiantes.

En la Universidad Laica Eloy Alfaro de Manabí, Extensión El Carmen, el proceso de planificación de horarios para la carrera de ingeniería en software suele hacerse manualmente, empleando papel, hojas de cálculo o documentos físicos, esto consume tiempo y causa errores humanos frecuentes; además, complica la ejecución de modificaciones súbitas a causa de sucesos inesperados; así mismo, la falta de un sistema centralizado complica la comprobación de todos los cambios implementados y limita que los profesores consulten horarios a tiempo.

El problema mencionado anteriormente da lugar a la necesidad de un enfoque tecnológico para proporcionar automatización y optimización de la programación para el área académica, brindando precisión, eficiencia y accesibilidad. Así, el proyecto propuesto aquí busca desarrollar una aplicación web que permita la planificación completa de horarios académicos con generación de horarios automática y manual, detección de conflictos, control de acceso basado en roles, gestión de información académica, generación de informes, entre otras funcionalidades. Esta aplicación web no solo mejorará la elaboración de los horarios académicos actuales, sino también agilizará un enfoque escalable por parte de la gestión educativa para la institución y fomentará el uso de las tecnologías de la información TI, como un medio estratégico para lograr la mejora continua de los procesos educativos. De esta manera, el proyecto se presenta como una alternativa viable y eficiente que aborda las circunstancias presentes y futuras en la carrera de ingeniería en software en ULEAM Extensión El Carmen.

1.2 Presentación del tema

Aplicación web para la planificación académica de horarios en la carrera de ingeniería en Software en ULEAM Extensión El Carmen.

1.3 Ubicación y contextualización de la problemática

Universidad Laica Eloy Alfaro de Manabí (ULEAM), Extensión El Carmen, ubicada en el Cantón El Carmen, provincia de Manabí, Ecuador, en la Avenida 3 de Julio cerca del cuerpo de bomberos del Cantón. En la actual planificación académica educativa, el esfuerzo por llegar a coordinar todos los horarios de la carrera de software manualmente se está convirtiendo en un problema común a medida que la población estudiantil se expande y el plan de estudios se aumenta periódicamente, y así los espacios físicos y los recursos de los instructores son limitados. La ausencia de alguna herramienta tecnológica en particular para la planificación de horarios lleva a la falta de organización, la efectividad administrativa y la satisfacción de los docentes de la carrera de ingeniería en software.

Por otro lado, la carrera de ingeniería en software requiere una programación precisa para que los recursos se usen eficazmente y se ofrezca una educación de alta calidad, sin embargo, el enfoque vigente no es lo bastante flexible para un entorno que es cada vez más complejo y dinámico, esto pone de manifiesto la necesidad de emplear tecnologías y aplicaciones web en la carrera para modernizar el proceso.

1.4 Planteamiento del problema

1.4.1 Problematización

En la actualidad, la implementación de horarios en la carrera de Ingeniería de Software se realiza manualmente, lo que resulta en la presencia de varias dificultades como asignaciones incorrectas de aulas, superposición de materias, sobrecarga de horarios para los profesores y uso ineficiente del tiempo disponible. Estas deficiencias influyen directamente en el proceso de enseñanza-aprendizaje, causando insatisfacción en estudiantes y profesores, y retrasos en el inicio de las actividades académicas.

El personal que proporciona un plan (hasta ahora) pasa mucho tiempo revisando, corrigiendo y ajustando los horarios manualmente, y no tienen un sistema disponible que pueda automatizar algunas tareas o detectar estos errores temprano en el proceso. Esto no solo ralentiza el proceso, sino que también lo hace propenso a errores humanos. Además, la ausencia de una base de datos sistemática o un historial digital para acceder o comparar la

actividad programada de los horarios de semestres anteriores dificulta la posibilidad de mejora y el patrón de uso en aulas y hábitos de los profesores.

Tal falta de fuentes organizadas es un obstáculo para la toma de decisiones y baja eficiencia para la función administrativa. Esto empeora a medida que aumenta la matrícula, donde hay un tremendo aumento en términos de combinaciones potenciales entre materias, profesores, grupos y aulas, y donde esto se volvería insostenible para la planificación manual, a mediano plazo. Por lo tanto, en este sentido, la necesidad de introducir una solución tecnológica que apoye la programación automática, segura y precisa y la optimización de la distribución es evidente.

1.4.2 Génesis del problema

Actualmente, no existe un sistema tecnológico especializado para la planificación automática y eficiente de horarios académicos en la Extensión El Carmen. Hasta ahora, se han utilizado métodos manuales basados en hojas de cálculo y documentos físicos, adecuados en los primeros años debido a la baja matrícula y oferta académica limitada. Sin embargo, el crecimiento significativo de estudiantes, asignaturas y docentes ha incrementado la complejidad organizativa, haciendo obsoletas estas prácticas tradicionales que no han evolucionado para integrar herramientas digitales ni un sistema unificado que centralice la información clave.

La escasez de actualización ha provocado problemas entre los que se incluyen conflictos entre aulas y profesores, errores en la distribución de horarios y una ardua responsabilidad administrativa, además, no contar con registros históricos digitalizados hace imposible detectar y corregir patrones negativos que se repiten aunque la situación se deteriora cada vez más por la ausencia de procesos de mejora continua, por lo tanto las dificultades aparecen cada semestre, no hemos implementado propuestas ni procedimientos formales para evaluar y optimizar el sistema de planificación actual de horarios, lo que muestra una resistencia o debilidad en el uso de tecnologías en la carrera.

1.4.3 Estado actual del problema

Actualmente, la carrera enfrenta problemas persistentes debido a una planificación de horarios poco eficiente. Conflictos en la asignación de espacios, sobrecarga de trabajo para los docentes y dificultades para coordinar asignaturas obligatorias son situaciones comunes cada semestre. Aunque se reconoce la necesidad de mejorar este proceso, aún no se han

incorporado herramientas digitales específicas que ofrezcan una solución integral adaptada a las características particulares de la Extensión El Carmen.

Gran parte del problema radica en que la elaboración de horarios se realiza de forma manual, lo que implica que el personal debe organizar una gran cantidad de datos, como materias, paralelos, docentes, bloques horarios y aulas disponibles, sin un soporte tecnológico que relacione automáticamente estas variables. Esto hace que el proceso sea laborioso, propenso a errores y a imprecisiones, obligando a realizar cambios de último momento. Esto provoca que un docente quede asignado a dos clases en el mismo horario de la misma jornada en diferentes niveles, evidenciando la falta de verificación automática para prevenir los conflictos en la planificación académica de horarios.

Otro inconveniente se puede señalar que el uso inadecuado de las aulas también afecta la organización académica, como la asignación de espacios no disponibles o inadecuados para ciertas asignaturas como por ejemplo utilizar el laboratorio para una clase teórica y viceversa, lo que provoca una nueva distribución en los horarios generando pérdida de tiempo. La falta de un sistema de planificación web dificulta el control de estos aspectos y genera múltiples solicitudes de ajuste por parte del encargado en la distribución de los horarios. Asimismo, la falta de una base de datos histórica impide el análisis y la mejora continua, ya que no se puede hacer un seguimiento de los errores ni optimizar la distribución docente con información previa, obligando a comenzar de cero cada semestre.

1.5 Diagrama causa – efecto del problema

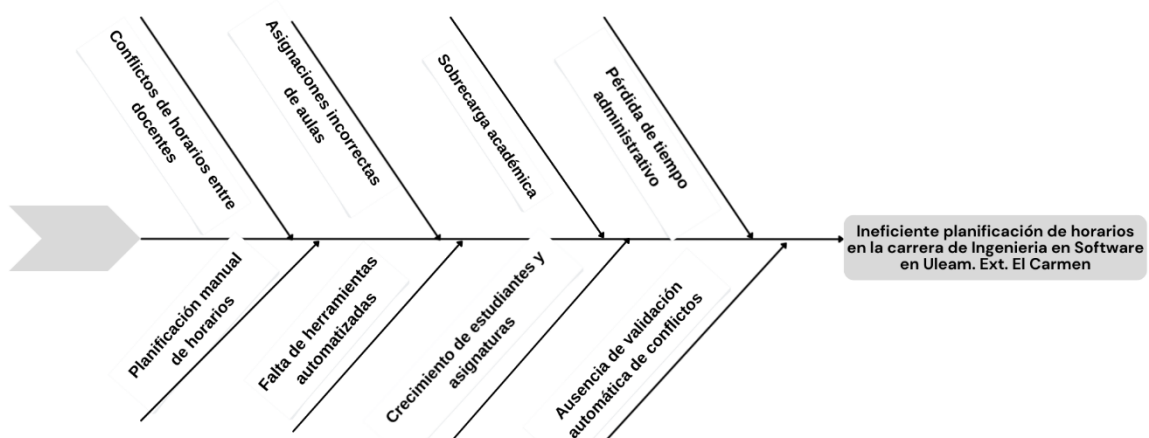


Ilustración 1: Diagrama de causa-efecto

1.6 Objetivos

1.6.1 Objetivo general

Desarrollar una aplicación web para la planificación académica de horarios en la carrera de ingeniería en Software en ULEAM Extensión El Carmen.

1.6.2 Objetivos específicos

- Investigar referencias bibliográficas sobre sistemas de aplicación web e implementación académica para sustentar la propuesta del proyecto.
- Analizar el procedimiento actual de programación de horarios académicos en la carrera de Ingeniería en Software de la ULEAM, para detectar los problemas y requerimientos más relevantes.
- Diseñar una aplicación web que a través de funcionalidades automatizadas y una interfaz intuitiva facilite la planificación académica de horarios y disminuya los errores.
- Implementar la aplicación web de los horarios y asegurar que esté en funcionamiento y así sea fácil de usar y se adapte a los retos académicos de la carrera de Ingeniería en Software.
- Validar la efectividad de la herramienta desarrollada a través de encuestas y entrevistas con los usuarios administrativos y docentes.

1.6.3. Hipótesis

La implementación de una aplicación web permitirá optimizar el proceso de elaboración de horarios académicos en la carrera de Ingeniería en Software de la ULEAM, reduciendo los errores humanos, mejorando la organización institucional y facilitando el acceso a los docentes.

1.7 Justificación

La planificación académica de horarios constituye un proceso esencial para agilizar la eficiencia en la implementación de los horarios, en la carrera de ingeniería en software de la Universidad Laica Eloy Alfaro de Manabí - extensión El Carmen, la elaboración de los horarios se realiza manualmente lo que genera dificultades como conflictos, retraso en la elaboración, pérdida de tiempo y otros problemas, por lo tanto esta situación ha evidenciado la necesidad urgente de implementar una herramienta tecnológica que automatice y optimice dichos procesos de planificación académica.

La propuesta de desarrollar una aplicación web se justifica plenamente en función de los beneficios que pueden ser de utilidad en la reducción de errores, ahorro de tiempo, automatización de tareas repetitivas y una mejor asignación de recursos humanos y físicos. Al contar con una plataforma centralizada que permita la gestión académica de los horarios, niveles, paralelos, aulas y asignaturas, se facilita la distribución entre las áreas académicas, lo cual se traduce en una mejora significativa del servicio ofrecido para los docentes.

Además, esta solución impulsa el principio de modernización institucional, al integrar herramientas digitales en los procesos de planificación de horarios. La automatización del sistema permitirá realizar ajustes en tiempo real, almacenar información histórica útil para futuras planificaciones o consultas, además, contribuirá con una gestión más ordenada, transparente y coherente con los objetivos estratégicos de la carrera.

El desarrollo de esta aplicación se alinea perfectamente con el perfil profesional de la carrera de Ingeniería en Software, por lo tanto, permite aplicar conocimientos adquiridos en asignaturas como ingeniería de software, bases de datos, programación web y análisis de sistemas, así, fomenta el aprendizaje significativo, la resolución de problemas reales del entorno académico y el compromiso en el uso de nuevas tecnologías.

Ante todo, lo mencionado, este proyecto no solo busca resolver conflictos en la planificación de horarios académicos, sino también convertirse en una propuesta escalable a futuro que pueda ser adaptada a otras carreras, extensiones o instituciones que tengan similares conflictos, entonces el impacto genera un cambio positivo en la gestión educativa mediante el uso de tecnologías emergentes.

1.8 Impactos esperados

1.8.1 Impacto tecnológico

El presente proyecto representa un avance significativo en el uso de herramientas tecnológicas dentro del ámbito universitario, ya que permitirá introducir una solución digital adaptada a las necesidades específicas de la planificación académica, en la actualidad, muchos procesos dentro de la ULEAM, Extensión El Carmen, aún se realizan de manera manual o con recursos digitales limitados, lo que impide alcanzar niveles óptimos de eficiencia, el objetivo de desarrollar la aplicación web es modificar uno de los elementos fundamentales de la gestión académica que es el diseño preciso de los horarios.

Se podrá observar en múltiples niveles el impacto tecnológico de esta solución, primeramente, se automatizarán las labores cotidianas y susceptibles a errores humanos, como

la designación de materias, aulas y horarios, además de optimizar la exactitud en la creación de los horarios, esta automatización también hará posible que se ahorre tiempo y recursos así mismo, este proyecto promueve la cultura tecnológica de la universidad al demostrar cómo el desarrollo de software puede solucionar problemas concretos en el ambiente educativo para la carrera de Ingeniería en Software, este tipo de aplicación se convierte en un referente de buenas prácticas académicas, ya que muestra cómo el conocimiento técnico adquirido en el aula puede tener un impacto directo en la mejora de los procesos internos, incluso puede estimular a estudiantes y docentes a participar en otros proyectos de innovación institucional que reflejen el aprendizaje estudiantil.

Se demuestra un potencial de escalabilidad y adaptación de la herramienta, aunque inicialmente se desarrollará específicamente para la carrera de Ingeniería en Software, pero se puede desarrollar desde un punto de vista de diseño para extenderse a otras carreras, extensiones o facultades dentro de la ULEAM. Esto significa que el proyecto no solo aborda una necesidad específica, sino que también se convierte en una propuesta tecnológica capaz de alcanzar una mayor difusión dentro de la institución educativa superior.

1.8.2 Impacto social

La aplicación web traerá beneficios para estudiantes, profesores y personal administrativo y mejorará la calidad de la vida académica en las universidades, la imagen de la universidad como una institución moderna, inteligente y eficiente se verá reforzada al inicio de cada ciclo académico, la reducción del estrés y la incertidumbre será uno de los beneficios sociales más cruciales, actualmente muchos estudiantes tienen cursos superpuestos cambios de horario y asignaciones de aulas ambiguas que contribuyen a la insatisfacción y desorganizan el tiempo personal y familiar.

La nueva aplicación web está destinada a ofrecer una planificación mejorada y más fácil, que sea más clara, accesible y estable, mejorando la experiencia educativa y estableciendo un modelo organizativo más sistemático o estructurado. Desde un punto de vista institucional, este proyecto puede ayudar a elevar el perfil público de la ULEAM en la población general y específicamente dentro de la Extensión El Carmen. El funcionamiento interno, la gestión de la tecnología y las tecnologías educativas que se utilizan mejoran la confianza entre los estudiantes actuales y futuros, los padres, las autoridades locales y otros interesados en la educación. Esto podría hacer que la universidad sea más prestigiosa y ayudar en su matrícula y retención de estudiantes.

1.8.3 Impacto ecológico

Al disminuir el uso de documentos de papel impresos para la planificación de horarios, el proyecto aportará a la reducción del consumo de papel, promoviendo prácticas de mejoramiento y empatía más sostenibles y amigables con el medio ambiente.

Con la implementación de una aplicación web, todo el proceso de planificación, distribución y comunicación de horarios se realizará de manera digital. Los horarios podrán consultarse en línea, visualizarse desde dispositivos móviles o descargarse en formato electrónico, lo cual eliminará la necesidad de imprimir múltiples versiones de cada horario para su revisión, difusión o corrección. Esta transformación no solo representa una mejora en la eficiencia del proceso, sino también una clara reducción del impacto ambiental derivado del uso de papel.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes históricos

La tecnología educativa ha evolucionado a lo largo de los años, pasando de técnicas manuales a modos de operación automatizados y digitales. El proceso de planificación académica al principio se ejecutaba completamente de manera analógica, con formularios físicos y procesos artesanales que limitaban la capacidad de responder a cambios dinámicos en el entorno universitario (Barragán, 2021). Esta situación duró hasta el avance tecnológico, lo cual impulsó a las universidades a utilizar la tecnología para optimizar su gestión académica.

Existe el caso de las aplicaciones web, que van a señalar un cambio en este contexto. Con la aparición de Internet en la década de 1990 y su posterior consolidación como plataforma de servicios, muchas universidades pudieron comenzar a digitalizar sus procesos internos, por ejemplo, la programación académica, y habilitar sistemas centralizados disponibles desde cualquier lugar con un punto de conexión (Méndez, 2022).

La agilidad administrativa se vio transformada por el cambio y la experiencia dinámica se volvió más sencilla para los docentes. Según Díaz (2023), en su artículo sobre educación superior al incorporar soluciones digitales la planificación institucional y sus metas educativas se integran de manera más cercana y estas herramientas ofrecen facilidades para crear modelos académicos de mayor envergadura, que son capaces de ajustarse a las exigencias específicas del mercado laboral y a los avances en tecnologías educativas, además, el uso de aplicaciones web para la planificación permite responder con mayor rapidez ante situaciones inesperadas, como crisis sanitarias o cambios de políticas educativas.

Por su parte, Castillo (2020) sostiene que la transformación digital no solo optimiza procesos, sino que también cambia la lógica organizacional de las universidades, impulsando modelos más ágiles, colaborativos y centrados en la toma de decisiones basadas en datos. La planificación académica, en este marco, se convierte en una estrategia estructural clave, y no meramente operativa, que impacta directamente en la calidad y pertinencia de los programas académicos.

2.2 Antecedentes de investigaciones relacionadas al tema presentado

Debido al acelerado avance científico y tecnológico, las universidades se han visto forzadas a modernizar su gestión administrativa y académica para alcanzar mayores niveles de eficiencia y excelencia, estas entidades, que no solo producen conocimiento, sino que también lo distribuyen, desempeñan un papel fundamental en el avance social al facilitar la optimización de procesos y expandir su influencia mediante la incorporación de tecnologías, para que en este sentido, la introducción de aplicaciones web como la propuesta para organizar los horarios académicos en la carrera de Ingeniería en Software se presenta como una solución innovadora que agilizará los procedimientos de planificación de horarios para enfrentar los desafíos presentados en los conflictos (Bernal y otros, 2024).

De igual manera, en la Universidad Técnica del Norte, Peralta y Mendoza (2022) desarrollaron una aplicación web con arquitectura MVC destinada a gestionar horarios académicos en programas técnicos. La investigación indicó que dicho sistema favoreció una distribución más justa de las cargas en los docentes y una gestión más eficiente de las aulas disponibles, gracias a la implementación de algoritmos que prevenían conflictos en los horarios.

Un análisis realizado en la Universidad Politécnica Salesiana por Herrera y López (2023) subrayó que los sistemas web destinados a la planificación académica deben tener en cuenta no solo las horas asignadas, sino también otros factores como la disponibilidad del personal como docentes, las infraestructuras y las normativas institucionales para esta investigación se recomienda utilizar tecnologías que permitan el acceso en la nube y desde dispositivos móviles mejorando la accesibilidad.

2.3 Definiciones conceptuales

2.3.1 Aplicación web.

Según Guevara (2022), se trata de un programa informático destinado a la elaboración, edición y procesamiento mediante el uso de computadoras o dispositivos móviles, además de que las aplicaciones web son desarrollos únicos que se crean para automatizar cualquier software según los requerimientos de un usuario, estas aplicaciones poseen una interfaz adaptada, creada según las necesidades específicas y la funcionalidad requerida que se ejecutan en un servidor a través de un navegador con conexión a internet, lo que elimina la necesidad de instalación en un dispositivo para su uso, con esto facilita el acceso al portal desde cualquier lugar y en cualquier momento.

El propósito de este estudio es examinar la cantidad, calidad y diversidad de aplicaciones web desarrolladas en Java y Python en diferentes sectores, identificando las ventajas particulares de cada lenguaje. También se analizan las diferencias clave entre ambos en términos de sintaxis, paradigmas de programación, herramientas de desarrollo y su popularidad actual entre los desarrolladores, considerando tendencias y preferencias dentro de la comunidad. Además, se evalúa cómo estas discrepancias influyen en aspectos críticos como la velocidad de desarrollo, escalabilidad y mantenimiento de las aplicaciones. Por último, se compara el rendimiento relativo de ambos lenguajes de programación respecto a la eficiencia y velocidad de ejecución para determinar cuál es más adecuado en diversos contextos del desarrollo web (Ecler & Luz, 2024).

2.3.1.1 Diseño de una aplicación web.

Las aplicaciones web están diseñadas siguiendo una arquitectura cliente-servidor. El cliente (navegador) solicita información, el servidor la procesa y devuelve los resultados (Silva y Calderón, 2023). Esta arquitectura se compone de tres capas principales.

- **Capa de presentación:** desarrollada con tecnologías como HTML, CSS y JavaScript, encargada de la interfaz del usuario.
- **Capa lógica o de negocio:** implementada con lenguajes como PHP, Python, Java o Node.js. Gestiona las operaciones y reglas del sistema.
- **Capa de datos:** que administra el almacenamiento, consulta y actualización de la información, comúnmente utilizando sistemas gestores de bases de datos como MySQL, PostgreSQL o MongoDB.

Este modelo de diseño modular facilita el mantenimiento, la escalabilidad y la seguridad del sistema, facilitando las actualizaciones a nuevas versiones según el avance tecnológico.

2.3.1.2 Evolución de las aplicaciones web

Existen actualmente frameworks modernos como React que permiten desarrollar aplicaciones altamente responsivas y personalizadas adaptadas a los requisitos del cliente, como menciona Rivera y Morales (2022), la evolución de la tecnología ha facilitado que las aplicaciones web compitan con las aplicaciones móviles en términos de rapidez en su rendimiento y lo intuitivo con el usuario especialmente debido a la aparición de las PWA (aplicaciones web progresivas). Desde los años 90, cuando eran sitios estáticos, las

aplicaciones web han avanzado hasta convertirse en sistemas altamente interactivos y complejos en la actualidad. En sus inicios, los sitios web solo ofrecían información básica; con la llegada de tecnologías como JavaScript, AJAX y HTML5, se dio paso a aplicaciones más dinámicas.

2.3.1.3 Procesos para diseñar una aplicación web

El desarrollo de una aplicación web requiere un conjunto de etapas metodológicas que van desde el análisis de requerimientos hasta su implementación y mantenimiento. De acuerdo con Gómez y Ríos (2021), el ciclo de vida del desarrollo de una aplicación web consta de:

1. Análisis de requerimientos funcionales y no funcionales

Es la etapa inicial donde se identifican y documentan las necesidades del usuario y del sistema. Los requerimientos funcionales indican lo que la aplicación debe hacer (por ejemplo: registrar usuarios, generar horarios), mientras que los no funcionales definen aspectos como el rendimiento, la seguridad y la usabilidad.

2. Diseño de la arquitectura y la base de datos

Consiste en definir la estructura técnica de la aplicación, es decir, cómo se organizarán los componentes (cliente-servidor, capas, servicios, etc.) y cómo se gestionará la información mediante el modelo de base de datos. Este diseño sirve como plano para el desarrollo.

3. Desarrollo del frontend y backend

Es la codificación de la aplicación. El frontend se refiere a la interfaz visual con la que interactúa el usuario para ello se utilizó un framework React JS, (diseñado con HTML, CSS, JavaScript, etc.), y el backend es la lógica del servidor (normalmente desarrollada en PHP, que procesa los datos y se comunica con la base de datos).

4. Pruebas funcionales de seguridad y usabilidad

Se verifica que la aplicación funcione correctamente y que las pruebas funcionales comprueban si cumple con los requerimientos, las de seguridad detectan vulnerabilidades, y las de usabilidad evalúan la experiencia del usuario en cuanto a facilidad de uso y comprensión de la interfaz.

5. Implementación y puesta en producción

Esta fase implica instalar la aplicación en un servidor real para que los usuarios finales puedan acceder a ella. También incluye configurar dominios, certificados de seguridad (SSL), bases de datos y realizar los últimos ajustes para asegurar su correcto funcionamiento en el entorno real.

6. Mantenimiento y actualizaciones continuas

Después del lanzamiento, es necesario realizar monitoreo, corrección de errores, mejoras funcionales y actualizaciones periódicas para mantener la aplicación segura, eficiente y alineada con los cambios tecnológicos o necesidades del usuario.

2.3.1.4 Tecnologías asociadas al desarrollo web

1. Frontend

El frontend o parte visible del sistema es la interfaz con la que interactúan los usuarios. Lenguajes como HTML5 y CSS3 son fundamentales para estructurar y dar estilo a las páginas web. Complementariamente, JavaScript permite dotar a la interfaz de interactividad, siendo potenciado por frameworks modernos como React, Angular y Vue.js. Estas tecnologías facilitan la creación de interfaces responsivas, accesibles desde diferentes dispositivos, lo cual es esencial en entornos universitarios donde los usuarios acceden desde portátiles, tabletas y teléfonos inteligentes (Zambrano y Vargas, 2024).

2. Backend

El backend es la lógica de programación que se ejecuta en el servidor. Aquí se procesan datos, se aplican reglas del negocio y se conecta con la base de datos. Tecnologías como PHP (frecuente en sistemas educativos como Moodle), Python mediante frameworks como Django o Flask, y Node.js (por su naturaleza asíncrona y rendimiento), son muy utilizadas. Además, lenguajes como Java con frameworks como Spring Boot ofrecen robustez y escalabilidad, características clave para aplicaciones institucionales (Cueva y Hernández, 2023).

3. Bases de datos

Toda aplicación web necesita una base de datos que almacene, organice y permita recuperar la información. MySQL y PostgreSQL son gestores de bases de datos relacionales ampliamente utilizados por su fiabilidad y compatibilidad con múltiples lenguajes de

programación. Por otro lado, MongoDB, como base de datos NoSQL, ofrece una estructura flexible basada en documentos, ideal para proyectos que manejan datos no estructurados o cambian constantemente (Mora y Castillo, 2022).

4. Control de versiones

El control de versiones es esencial para gestionar los cambios realizados en el código fuente. Herramientas como Git, junto con plataformas colaborativas como GitHub, GitLab o Bitbucket, permiten trabajar en equipo, mantener un historial de modificaciones y facilitar la integración continua. Estas herramientas son especialmente útiles en proyectos académicos donde intervienen varios desarrolladores o tutores que supervisan los avances (Silva y Román, 2021).

5. Sistemas de despliegue

Finalmente, el despliegue de una aplicación web requiere herramientas que faciliten su publicación, escalado y actualización. Tecnologías como Docker permiten contenerizar aplicaciones y ejecutarlas en cualquier entorno sin problemas de compatibilidad. Asimismo, la implementación de prácticas de Integración y Entrega Continua (CI/CD) automatiza los procesos de prueba y publicación del sistema. Servidores web como Apache o Nginx actúan como intermediarios para entregar los contenidos de la aplicación a los usuarios finales (García y Menéndez, 2024).

2.3.1.5 Ventajas y desafíos

Entre las ventajas de una aplicación web se encuentran la facilidad de acceso desde cualquier dispositivo, menor coste de mantenimiento, actualizaciones centralizadas y posibilidad de integración con otros sistemas. Sin embargo, también enfrentan desafíos como la seguridad ante ataques informáticos, tiempos de carga, y dependencia de la conectividad a internet (Peña y Delgado, 2023).

2.3.1.6 Planificación académica de horarios

Tradicionalmente, este tipo de planificación se realizaba de forma manual mediante hojas de cálculo o formatos impresos, lo cual demandaba una gran inversión de tiempo por parte del personal administrativo, además de estar expuesta a errores humanos, conflictos de horarios y una baja adaptabilidad frente a cambios imprevistos. La falta de automatización en la planificación puede generar problemas como sobrecarga de trabajo docente, solapamientos

entre clases, uso ineficiente de las aulas y desorganización general del semestre académico (Crespo y Palma, 2022).

Según Hernández Sampieri (2022), la planificación adecuada del calendario académico es esencial para el diseño de los horarios de la carrera, porque así permite la integración de los diversos elementos involucrados en la formación profesional de los docentes. La programación no debe ser vista simplemente como un procedimiento operativo, sino como un movimiento estratégico que influye en la calidad de la educación superior. A un nivel fundamental, la programación académica constituye uno de los procesos más relevantes dentro de la gestión universitaria, debido a que asegura el uso eficiente de los recursos institucionales, garantiza el cumplimiento de las debidas asignaciones y mejora la experiencia académica en docentes y parte de área administrativa. Este proceso, aunque tradicionalmente se ha desarrollado de forma manual, ha evolucionado gracias al uso de tecnologías informáticas, permitiendo mejoras significativas en la eficiencia, la equidad y la transparencia.

2.3.1.7 Características del proceso de planificación

El proceso de planificación académica de horarios se caracteriza por su multivariable complejidad. Incluye elementos como la disponibilidad de docentes, la asignación de aulas, el número de paralelos por materia, el uso de laboratorios, los requisitos previos de las asignaturas y las restricciones de tiempo y espacio. De acuerdo con Morales y Cabrera (2023), gestionar todos estos factores de forma manual conlleva un alto riesgo de error humano, por lo que se requiere la implementación de sistemas automatizados que optimicen este proceso.

2.3.2 Problemas frecuentes en la planificación manual

Uno de los principales desafíos en la planificación tradicional es la presencia de conflictos de horarios, duplicidad en la asignación de docentes o espacios, y la falta de flexibilidad para incorporar cambios de último momento. Estas fallas se deben, en parte, al uso de herramientas básicas como hojas de cálculo o formatos impresos, que carecen de algoritmos que validen las decisiones tomadas. Además, según Rivera y Llor (2020), este enfoque manual genera ineficiencia operativa, pérdida de tiempo administrativo y baja adaptabilidad frente a emergencias o ajustes curriculares.

2.3.2.1 Ventajas de la planificación automatizada

La automatización de la planificación académica permite superar muchas de las limitaciones mencionadas dado que los sistemas informáticos que utilizan algoritmos de

optimización y programación lógica son capaces de crear propuestas de horarios que optimizan el uso de recursos de la institución ayudando a reducir conflictos y proporcionan soluciones más rápidas, de acuerdo con Roldán y Espinoza (2023), estas aplicaciones web no solo facilitan este proceso de ayudar con los errores sino que también permiten incorporar funciones como el registro de modificaciones y funcionalidades, la visualización dinámica del horario y la creación de informes.

2.3.2.2 Perspectiva pedagógica de la planificación

Desde el enfoque pedagógico, la planificación de horarios también debe considerar los ritmos de aprendizaje, la carga horaria óptima y la distribución equilibrada de contenidos a lo largo del semestre. De acuerdo con Alvarado y Quintero (2021), un horario bien estructurado influye positivamente en la motivación del estudiante, reduce el ausentismo y favorece una experiencia académica más satisfactoria. Además, permite a los docentes planificar con antelación sus actividades, generar espacios de tutorías y fortalecer la relación pedagógica.

2.3.2.3 Componentes clave en la planificación académica

La planificación académica de horarios integra diversos elementos fundamentales que deben ser considerados de manera coordinada. Según Peña y Rodríguez (2022), estos componentes incluyen: la distribución de la carga horaria, la disponibilidad del personal docente, la capacidad instalada (aulas, laboratorios y equipos), las políticas institucionales, y el calendario académico oficial. La adecuada gestión de estos factores permite optimizar el uso de los recursos educativos, mejorar la organización semestral y garantizar una experiencia académica más eficiente tanto para estudiantes como para docentes.

2.3.2.4 Herramientas tecnológicas utilizadas

Actualmente existen plataformas especializadas en planificación académica, tanto propietarias como de código abierto. Herramientas como UNISOFT Horarios, ASISTE, Gradschedule, o integraciones de módulos como OpenSIS y Moodle permiten gestionar este proceso. Sin embargo, muchas universidades optan por el desarrollo de soluciones a medida, basadas en aplicaciones web que se ajusten a sus necesidades internas, como el caso de la propuesta planteada en este proyecto para la carrera de Ingeniería en Software de la ULEAM Extensión El Carmen. (Vásquez y Delgado, 2024)

2.3.2.5 Impacto en la gestión académica

Una buena planificación impacta en la reducción de gastos administrativos, mejora la percepción institucional, incrementa la eficiencia operativa y fortalece el cumplimiento

normativo del currículo. Además, según Salinas y Romero (2023), las instituciones que aplican tecnologías de planificación muestran mayor capacidad de adaptación ante situaciones de emergencia, como la pandemia de COVID-19, donde la programación híbrida y la flexibilidad fueron claves.

2.3.2.6 Planificación académica en las instituciones de educación superior

La planificación académica universitaria es un proceso clave para garantizar la calidad y eficiencia de la educación superior. En términos generales, los fundamentos teóricos de la planificación académica universitaria se basan en tres pilares (Oca P. Y., 2023):

1. **Planificación estratégica:** La planificación académica debe estar alineada con la misión, visión y objetivos estratégicos de la institución. Es decir, se trata de un proceso integral que busca integrar las diferentes áreas de la universidad en torno a un proyecto común.
2. **Gestión por resultados:** La planificación académica debe estar orientada hacia resultados concretos y medibles. Para ello, se utilizan indicadores que permiten evaluar el desempeño de la institución y ajustar el plan en función de los resultados obtenidos.
3. **Enfoque en el estudiante:** La planificación académica debe poner al estudiante en el centro del proceso educativo. Esto implica considerar sus necesidades, intereses y expectativas para ofrecer programas académicos de alta calidad y relevancia social.

2.3.2.7 Definición especializada del proceso

La planificación académica de horarios consiste en elaborar un cronograma estudiantil que articula asignaturas, docentes, espacios físicos y horas académicas según los requisitos curriculares y logísticos institucionales, así, este proceso permite asegurar la capacidad pedagógica, el uso eficiente de los recursos y la accesibilidad para todos los actores involucrados (Oca y otros, 2023). Por lo tanto, esta planificación garantiza que cada estudiante pueda cursar sus asignaturas sin solapamientos y acceda a espacios adecuados que mantengan un ritmo académico que promueva el equilibrio entre teoría y práctica, que es un aspecto clave en la formación profesional de Ingeniería en Software, donde se combinan clases teóricas y de laboratorio además de tutorías personalizadas.

2.3.2.8 Componentes clave

Los componentes esenciales de la planificación incluyen la carga horaria de cada asignatura, la disponibilidad de los docentes, la infraestructura académica (aulas, laboratorios y salas de tecnología), el marco regulatorio institucional y el calendario académico oficial (Peña & Rodríguez, 2022). Cada uno de estos elementos debe interactuar de forma dinámica:

- La carga horaria establece cuántas horas semanales necesita cada asignatura para cumplir con su contenido curricular.
- La disponibilidad docente considera las horas laborales, restricciones contractuales y compromisos previos.
- La infraestructura debe coordinarse con las actividades prácticas y teóricas, asegurando espacios adecuados.
- La normativa y el calendario institucional determinan fechas clave (inicio semestres, evaluaciones, feriados) que condicionan la disponibilidad real del tiempo.

2.3.2.9 Diseño de implementación en sistemas automatizados

El diseño de un sistema automatizado para la planificación académica debe incluir varios módulos clave, como reglas curriculares, lógica de asignación horaria, validación de restricciones y generación de escenarios alternativos (Gómez, 2021). Estos módulos permiten:

- Definir reglas específicas (ej., disponibilidad docente, obligatoriedad de ciertos horarios en laboratorios).
- Automatizar la asignación inicial y ajustar iterativamente cuando se detectan solapamientos.
- Generar diferentes escenarios de horario, permitiendo comparaciones antes de tomar decisiones finales.
- Implementar un repositorio histórico de versiones, útil para auditoría y respuesta a imprevistos.

2.3.3 Implementación tecnológica y módulos

Un sistema de planificación debe incluir al menos los siguientes módulos (Ruben, 2022):

1. Registro docente, con datos personales, horarios disponibles y competencias asignadas.

2. Base de datos curricular, con asignaturas, créditos, prerrequisitos y modalidad (teórica, práctica).
3. Base de datos de espacios físicos, con capacidad, equipamiento y tipo de uso (teoría, laboratorio).
4. Módulo de generación de horarios, que utiliza algoritmos para combinar variables y validar restricciones.
5. Módulo de validación, que gestiona conflictos y ofrece soluciones alternativas.
6. Módulo de reportes, donde se generan documentos institucionales y estadísticas para la toma de decisiones.

2.3.3.1 Seguridad en la implementación

La protección de datos es esencial en cualquier sistema académico. Según Smith (2023), una aplicación confiable debe garantizar:

- Control de acceso granular según roles (coordinador, docente).
- Cifrado en reposo y tránsito, especialmente para datos sensibles como sus credenciales o información personal.
- Registro de auditorías, donde se registran todas las acciones relevantes (modificaciones, visualizaciones).
- Autenticación segura, usando contraseñas robustas o autenticación multifactorial.

2.3.3.2 Procesos de validación y ajustes en tiempo real

Los sistemas modernos de planificación académica integran procesos de validación constantes durante la elaboración de los horarios. Según López y Carrasco (2023), estos procesos permiten al coordinador académico verificar automáticamente restricciones como solapamientos, excedentes de carga docente y disponibilidad de aulas. Gracias a algoritmos adaptativos, el sistema sugiere ajustes en tiempo real y genera versiones alternativas del horario, misma que puede ser validada por el personal administrativo y académico antes de su publicación definitiva.

2.3.3.3 Indicadores de rendimiento del horario académico

Para medir la efectividad de un horario, se utilizan indicadores tales como el porcentaje de aulas utilizadas, número de solapamientos detectados, nivel de satisfacción docente y porcentaje de planes curriculares completados. Hernández y Silva (2022) proponen que estos indicadores deben ser calculados automáticamente por el sistema cada fin de

semestre, lo que facilita el análisis de datos y la toma de decisiones estratégicas para la mejora continua del proceso.

2.3.3.4 Integración con otros sistemas institucionales

Una aplicación web de planificación debe interoperar con plataformas como el sistema de información estudiantil, gestión de matrículas o bibliotecas digitales. Según Castro y Montenegro (2021), la integración mediante servicios web RESTFULL y APIS permite actualizar datos en tiempo real, evitando duplicaciones de registros y facilitando la sincronización intermodular. Además, centraliza la información institucional, favoreciendo vistas consolidadas y robusteciendo procesos transversales como la asignación de cargas docentes.

2.3.3.5 Escalabilidad y personalización del sistema

A medida que crecerá la matrícula estudiantil o se incluyen nuevas asignaturas, el sistema debe escalar sin perder rendimiento. Ramírez (2023), menciona que una solución basada en microservicios o contenedores permite adaptar módulos de planificación según la demanda. Asimismo, el sistema debe ofrecer personalización para carreras con requerimientos especiales, como Ingeniería en Software, donde existen laboratorios con disponibilidad limitada y módulos de horario diferenciados.

2.3.3.6 Formación del personal y adopción institucional

Aunque un sistema puede ser técnicamente robusto, su éxito depende de la capacitación del personal administrativo y académico. Pérez y Aguilar (2022), indican que los programas de formación deben estar incluidos en el diseño del sistema, con material tutorial, sesiones de práctica y manuales de usuario. Esta etapa es fundamental para reducir la resistencia al cambio, aumentar la precisión en la planificación y asegurar un uso continuo y correcto.

2.3.3.7 Simulación de escenarios académicos

La herramienta de planificación académica puede incluir una función de simulación que permite evaluar diferentes escenarios de horarios antes de seleccionarlos. Según Ortega y Suárez (2022), esto permite prever problemas como sobrecargas o solapamientos, facilitando la decisión sobre cuál escenario adoptar. Además, ayuda a los administradores a visualizar el impacto de cambios como incorporación de nuevos cursos o modificaciones en la matrícula.

2.3.3.8 Accesibilidad y experiencia del usuario (UX)

Un sistema de planificación eficaz no solo debe funcionar bien, sino también ser fácil de usar. Según González (2023), una interfaz diseñada con principios de experiencia de usuario reduce el tiempo de entrenamiento y mejora la satisfacción de docentes y coordinadores. Las buenas prácticas incluyen diseño responsivo, navegación clara y accesibilidad para personas con discapacidad, asegurando que la aplicación sea útil para todos los usuarios.

2.3.3.9 Gestión de incidencias planificadas

Durante el semestre pueden surgir situaciones imprevistas como clases extraordinarias o cancelaciones. El sistema debe incorporar un módulo de gestión de incidencias que permita registrar modificaciones, reasignar espacios y notificar automáticamente a los afectados. Martínez y Ruiz (2022) señalan que esta capacidad es esencial para asegurar la continuidad educativa y la flexibilidad operativa.

2.3.4 Auditoría y trazabilidad

Es fundamental poder auditar el historial de cambios en la planificación. Según Navarro y López (2023), un módulo de trazabilidad permite recuperar versiones anteriores, verificar modificaciones y asegurar transparencia. Esto asegura responsabilidad administrativa y ayuda en casos de auditoría interna o revisión curricular.

2.3.4.1 Gestión del capital humano docente

La planificación académica debe considerar no solo los horarios de clase, sino también la disponibilidad y carga horaria del personal docente. Según Cabrera y Arévalo (2021), una correcta gestión del talento humano es clave para mantener el equilibrio entre la calidad educativa y el bienestar de los profesores. Esta gestión implica asignaciones racionales, respeto por las horas contratadas y equilibrio entre grupos asignados.

2.3.4.2 Evaluación de la eficiencia horaria

Una buena planificación no se limita a cumplir con una malla curricular, sino que busca optimizar el uso del tiempo disponible. Salazar (2022) propone medir la eficiencia horaria como el porcentaje de horas efectivas de clase frente al total de horas asignadas, permitiendo detectar tiempos muertos, horas subutilizadas o excesos en ciertos periodos.

2.3.4.3 Automatización del proceso de planificación

Una de las principales ventajas de una aplicación web es la capacidad de automatizar procesos complejos. Según Hernández y López (2023), la automatización en la planificación

de horarios reduce hasta en un 70% el tiempo de trabajo administrativo, disminuye errores humanos y mejora la capacidad de respuesta ante cambios. Este tipo de automatización se logra mediante algoritmos de asignación y sistemas basados en reglas.

2.3.4.4 Seguridad y control de acceso en la planificación

La información relacionada con horarios, docentes y grupos académicos debe estar protegida para evitar modificaciones no autorizadas. Según Morales (2021), los sistemas de planificación académica deben contar con mecanismos de autenticación, control de permisos y trazabilidad de cambios. Además, se recomienda cifrado de datos sensibles y respaldo automático para garantizar la integridad de la información.

2.3.4.5 Integración con sistemas institucionales

La planificación académica de horarios no debe trabajarse de forma aislada, sino integrada con otros sistemas como el de matrícula, evaluación docente y biblioteca. Según Jiménez y Vargas (2024), una correcta integración reduce la duplicidad de datos y mejora la sincronización entre procesos académicos, elevando la eficiencia operativa de la institución.

2.3.4.6 La planificación académica como proceso estratégico para la sostenibilidad institucional

La planificación académica no solo organiza el presente, sino que orienta el futuro institucional. De acuerdo con Ortega y Díaz (2022), al establecer cronogramas, cargas docentes, secuencias curriculares y disponibilidad de aulas, se crea una hoja de ruta que permite a la universidad anticiparse a retos como cambios en la demanda estudiantil o escasez de recursos. En este sentido, la planificación académica de horarios se convierte en un pilar estratégico para la sostenibilidad y crecimiento de la educación superior.

2.3.4.7 Planificación académica orientada al estudiante mediante TIC

La planificación horaria debe estar centrada en el estudiante, permitiéndole acceder a una oferta académica diversa, organizada y coherente. Según Roldán (2023), las TIC, y en especial las aplicaciones web, pueden facilitar esa orientación al estudiante al brindar interfaces amigables donde visualice horarios, evite solapamientos y elija turnos según sus necesidades. Esto mejora la experiencia académica y fortalece la retención estudiantil.

2.3.5 Metodología propuesta

Modelo en Cascada

El desarrollo de sistemas de software se apoya en metodologías que estructuran y organizan las etapas necesarias para entregar productos funcionales. Una de las metodologías

más utilizadas en proyectos académicos e institucionales es el modelo en cascada, también conocido como Waterfall. Este modelo representa un enfoque secuencial en línea donde cada fase del desarrollo debe completarse en su totalidad antes de pasar a la siguiente, lo que permite un mayor control en todos los procesos (Schach, 2022).

2.3.5.1 Definición y características del modelo en cascada

El modelo en cascada fue uno de los primeros enfoques sistemáticos para el desarrollo de software. Su estructura divide el proceso en etapas bien definidas: análisis, diseño, implementación, pruebas, despliegue y mantenimiento. Cada una de estas etapas tiene objetivos, productos y entregables específicos, lo que permite llevar un control riguroso del progreso del proyecto. Esta metodología se recomienda especialmente cuando los requisitos del sistema son claros desde el inicio, como en el presente caso, donde el objetivo es construir una aplicación web con funcionalidades delimitadas para la planificación académica de horarios.

El modelo en cascada está compuesto por una serie de etapas secuenciales: análisis, diseño, implementación, pruebas, despliegue y mantenimiento. Según Pressman y Maxim (2021), cada una de estas fases tiene entregables definidos y se construye sobre la salida de la fase anterior, lo cual reduce la ambigüedad y mejora la planificación de tareas. Esta metodología se considera apropiada para proyectos individuales como este, donde el alcance es claro, los recursos son limitados y se requiere precisión técnica en cada entrega.

2.3.5.2 Fases del modelo en cascada aplicadas al proyecto

1. Análisis de requerimientos

En esta etapa se identifican las necesidades funcionales y no funcionales del sistema, tales como la gestión de la carga docente, el registro de asignaturas, la disponibilidad de aulas, y la generación automática de horarios por semestre. De acuerdo con Bonilla (2020), esta etapa es crucial, ya que los errores en el levantamiento de requisitos pueden tener consecuencias graves en las fases posteriores, especialmente en modelos rígidos como el modelo en cascada.

2. Diseño del sistema

Una vez establecidos los requerimientos, se procede a la estructuración del sistema desde una perspectiva técnica. Según Escobar (2021), un diseño bien elaborado reduce el

riesgo de fallos y optimiza el rendimiento del sistema, lo que es clave en herramientas que apoyan la gestión institucional.

3. Implementación

En esta fase se procede a la codificación de la aplicación, transformando el diseño técnico en componentes funcionales. Según Gutiérrez (2023), una implementación bien planificada facilita la adopción del sistema y reduce los errores por parte de los usuarios, aspecto vital en aplicaciones educativas.

4. Pruebas

Las pruebas permiten verificar que cada componente cumple con los requerimientos establecidos y funciona correctamente bajo diferentes condiciones. Hernández y Rincón (2023) sostienen que una fase de pruebas rigurosa es indispensable para asegurar la calidad del software y garantizar una experiencia de usuario adecuada en entornos reales.

5. Despliegue

El sistema será desplegado en un entorno de prueba dentro del ámbito universitario para ser evaluado por los usuarios finales. Como explican Olarte y Rodríguez (2022), una buena planificación del despliegue permite que el sistema esté accesible, operativo y funcional para ser evaluado como prototipo institucional.

6. Mantenimiento

Aunque el alcance del proyecto no contempla una etapa prolongada de mantenimiento, se documentará el procedimiento para futuras actualizaciones o adaptaciones institucionales. De acuerdo con Sommerville (2020), el mantenimiento es un componente esencial en el ciclo de vida del software, incluso cuando se plantea como una fase de documentación futura.

2.3.5.3 Justificación del modelo en cascada según el contexto del proyecto

El presente proyecto es individual y tiene requerimientos fijos, definidos por las necesidades de la carrera de Ingeniería en Software de la ULEAM Extensión El Carmen. Por ello, el modelo en cascada se adapta correctamente, permitiendo cumplir con las exigencias institucionales, generar documentación sólida y establecer un camino predecible hacia la entrega del sistema. Pressman y Maxim (2021) destacan que este modelo sigue siendo uno de los más apropiados cuando el control del flujo del proyecto es más importante que la flexibilidad. Asimismo, el enfoque secuencial de este modelo evita los constantes cambios

propios de metodologías ágiles, lo cual es conveniente en entornos académicos donde el tiempo es limitado y las modificaciones en fases avanzadas pueden comprometer la entrega. Su claridad metodológica permitirá cumplir con los objetivos específicos planteados y facilitará el proceso de validación del producto final ante la comisión evaluadora.

2.4 Conclusiones relacionadas al marco teórico en referencia al tema planteado.

En primer lugar, la aplicación web se identificó como una solución tecnológica genial porque es accesible, flexible y permite mantener el control de la información en un solo lugar. Esto es súper importante para cualquier universidad porque ayudan a hacer más fáciles esas tareas administrativas que a veces presentan complicaciones, mejorando así la calidad de la educación, estas ventajas son clave para controlar mejor las materias, los horarios, los docentes y las aulas.

En segundo lugar, al examinar la organización de los horarios se reconoce que constituye un proceso integral que incide directamente en el uso de los recursos, en la calidad educativa y en la satisfacción del estudiantado. Una planificación adecuada permite equilibrar la carga laboral docente y optimizar la utilización de los espacios, previniendo inconvenientes operativos, lo cual resulta especialmente relevante en instituciones con alta población estudiantil como la ULEAM.

Al revisar libros y artículos sobre modelos de desarrollo de software, se identificaron diversas alternativas metodológicas y se determinó que el modelo en cascada resultaba el más adecuado para el proyecto. Aunque existen metodologías como Scrum orientadas al trabajo colaborativo, su enfoque no se ajusta a las características de un desarrollo individual. En este contexto, el modelo en cascada se considera apropiado cuando los requerimientos están claramente definidos desde el inicio.

Además, se comprendió que usar metodologías formales en el desarrollo de software no solo garantiza el correcto funcionamiento del sistema, sino que también favorece la transparencia y comprensión del proyecto, fortaleciendo la credibilidad de los resultados y facilitando su posible implementación institucional.

Finalmente, las soluciones tecnológicas deben adaptarse al contexto en el que serán utilizadas, considerando las necesidades de la institución, su dinámica de trabajo y sus procesos internos. Para que un sistema académico resulte efectivo, no basta con su calidad técnica, sino que también debe integrarse adecuadamente al entorno organizacional donde será aplicado.

CAPÍTULO III

MARCO INVESTIGATIVO

3.1 Introducción

Este capítulo presenta el marco metodológico que fundamenta la investigación para desarrollar una aplicación web de planificación de horarios académicos en la carrera de Ingeniería de Software de la ULEAM - Extensión El Carmen, detallando el tipo de estudio, métodos, fuentes e instrumentos de recolección y análisis de datos. Según Hernández y Baptista (2021), una adecuada metodología orienta el proceso investigativo, asegurando la calidad y validez de los resultados obtenidos.

3.2 Tipos de investigación

3.2.1 Investigación bibliográfica

El término investigar implica buscar y analizar para descubrir conocimiento, y en este proyecto, la investigación bibliográfica fue clave para construir el marco teórico y metodológico (Révolo-Acevedo et al., 2023, p. 13).

Se consultaron fuentes especializadas en desarrollo web, metodologías de software, gestión académica y planificación de horarios, lo que permitió fundamentar las variables de estudio y decisiones técnicas. Este análisis documental facilitó comparar enfoques, identificar tendencias en educación superior y seleccionar herramientas adecuadas, asegurando que la propuesta tecnológica se basara en fundamentos científicos y experiencias previas exitosas

3.2.2 Investigación de campo

Esta información permitió desarrollar una comprensión precisa y contemporánea del problema, así como validar los requisitos necesarios para el diseño de la aplicación web propuesta señalando que este estudio se realiza en el contexto donde suceden los fenómenos de interés, facilitando su análisis en un entorno auténtico (Sarango et al., 2024).

En el contexto de este proyecto, se implementó una investigación de campo con el objetivo de identificar las principales dificultades en la organización manual de horarios y recopilar recomendaciones sobre las funcionalidades más valiosas que debería tener una herramienta tecnológica.

3.3 Enfoques y métodos de la investigación

Los enfoques y métodos utilizados son:

3.3.1 Enfoque cualitativo

El enfoque cualitativo, según Sánchez Castillo y Gómez Cano (2024), busca comprender fenómenos sociales mediante la recopilación detallada de datos cualitativos.

En este proyecto, se aplicó a través de entrevistas al coordinador de carrera para explorar sus experiencias y desafíos en la planificación manual de horarios, y mediante preguntas cerradas y de selección múltiple en encuestas a los docentes para captar opiniones profundas. Estos datos fueron esenciales para identificar requisitos funcionales como usabilidad, flexibilidad y adaptabilidad que no se evidencian en la información cuantitativa.

3.3.1.2 Enfoque cuantitativo

El enfoque cuantitativo, según Tapia Camargo (2020), se basa en la recolección y análisis de datos numéricos mediante instrumentos estructurados para describir o predecir fenómenos.

En este proyecto, se aplicó una encuesta cuantitativa a nueve docentes de jornada completa con preguntas cerradas que midieron la frecuencia de conflictos de horario, la eficiencia del proceso actual y la aceptación de una aplicación web. Los datos obtenidos permitieron identificar patrones comunes y priorizar los requisitos funcionales y no funcionales de la aplicación, enfocándose en resolver los problemas más urgentes para la mayoría.

3.3.1.3 Método deductivo:

El método deductivo parte de principios generales para llegar a conclusiones específicas, confirmando hipótesis en contextos concretos y siendo útil cuando las causas de un fenómeno son difíciles de observar (Serrano, 2020).

En este proyecto, se aplicó este método basándose en principios sobre planificación académica y tecnologías, para analizar casos específicos en la carrera de Ingeniería de Software de la ULEAM, Extensión El Carmen.

3.3.1.4 Método analítico:

El método analítico permite descomponer un fenómeno en sus partes fundamentales para comprender su estructura y funcionamiento (Sierra Bravo, 2021).

Asimismo, se empleó el método analítico, el cual permite descomponer el proceso actual de planificación académica en sus partes esenciales como la asignación de aulas, horarios y docentes, para detectar con precisión sus debilidades y proponer una solución concreta a través del desarrollo de software.

3.4 Fuentes de información de datos

3.4.1 Fuentes primarias

Las fuentes primarias consisten en datos extraídos de participantes de la carrera de Ingeniería de Software en la Extensión El Carmen. Dichos participantes incluyen coordinador académico y docentes que nos proporcionan datos prácticos sobre el proceso tal como existe ahora, sobre sus deficiencias y sobre los cambios y esperanzas necesarios para digitalizar el proceso.

3.4.2 Fuentes secundarias

Las fuentes secundarias son aquellos textos bibliográficos, libros, investigaciones previas, manuales institucionales y documentos normativos que respaldan el plan teóricamente sólido para la planificación académica, la gestión de horarios y el desarrollo de aplicaciones web en contextos educativos. Estas fuentes científicas ofrecen apoyo para construir la base teórica del proyecto, así como su diseño metodológico y análisis de datos de los datos recopilados

3.5 Estrategia operacional para la recolección de datos

El propósito principal de esta investigación es adaptar y aplicar el instrumento de encuesta y entrevista para obtener un análisis descriptivo de los datos y poder realizar un diagnóstico de la comunidad en sus componentes sociales, demográficas, económicas y culturales, este análisis permitirá identificar áreas de mejora y fortalezas en la carrera (Lilián Falcón et al.,).

3.5.1 Población

La población objetivo de esta investigación está conformada por 9 docentes activos de la jornada completa y 1 coordinador académico de la carrera de Ingeniería en Software en la ULEAM, Extensión El Carmen, sumando un total de 10 participantes. Esta población fue seleccionada debido a su vínculo directo con el proceso de planificación académica de horarios, que constituye el objeto principal del presente estudio.

3.5.2 Segmentación

La población está segmentada de la siguiente manera:

- **Segmento 1:** Docentes de la carrera de Ingeniería en Software, quienes utilizan el sistema de horarios en sus actividades diarias y están directamente afectados por su eficiencia o ineficiencia.
- **Segmento 2:** Coordinador académico, quien es el encargado de organizar y validar la planificación de horarios académicos.

3.5.3 Censo poblacional.

Para la presente investigación no se aplicó un muestreo, debido a que se trabajó con la totalidad de la población identificada. En este sentido, el estudio se desarrolló bajo un censo poblacional, incluyendo a todos los docentes activos y al coordinador académico de la carrera de Ingeniería en Software en la ULEAM, Extensión El Carmen (Hernández Sampieri & Mendoza Torres, 2021).

La decisión de emplear un censo se fundamenta en el tamaño reducido de la población y en la necesidad de obtener información completa y representativa del proceso de planificación académica de horarios, evitando sesgos derivados de la selección parcial de participantes. Por lo tanto, la población y los participantes del estudio coinciden, quedando conformada por 9 docentes y 1 coordinador académico, para un total de 10 participantes.

3.6 Análisis de las herramientas de recolección de datos a utilizar

La evaluación y selección de herramientas para la recolección de información es un aspecto clave en los proyectos que requieren recopilación de datos y para asegurar la calidad de los datos, analizamos cuidadosamente los instrumentos de recolección identificando su idoneidad, confiabilidad y validez para nuestro estudio en particular, por ejemplo, la encuesta nos permitiría recopilar datos numéricos y estadísticos, mientras que la entrevista sería ideal para identificar plenamente las dificultades que se encuentran en la planificación actual, entonces se determina que estas herramientas son las más adecuadas para obtener datos precisos y representativos, teniendo en cuenta las particularidades de nuestra investigación. Tradicionalmente, uno de los métodos más comunes es el cuestionario, empleado tanto en investigaciones cuantitativas como cualitativas, ya que facilita la obtención y registro de datos

mediante diversas preguntas sobre los aspectos relevantes del estudio, destacándose como una herramienta muy versátil (Cisneros et al., 2022).

3.6.1 Encuesta

Se elaboró un cuestionario dirigido a docentes, con preguntas cerradas que permitan obtener datos cuantitativos sobre el uso actual de herramientas de planificación, tiempos estimados de organización, frecuencia de errores y aceptación de nuevas tecnologías.

3.6.2 Entrevista

Según Ocaña-Fernández y Luján-Mora (2021), las entrevistas cualitativas de tipo abierto permiten acceder a información relevante que no se puede obtener mediante instrumentos cerrados, ya que brindan libertad al informante para expresar sus ideas, percepciones, experiencias y sugerencias de forma espontánea.

Para el desarrollo de este trabajo investigativo, se aplicó una entrevista al coordinador de la carrera de Ingeniería en Software en la ULEAM Extensión El Carmen. Esta técnica se seleccionó debido a que permite explorar en profundidad aspectos clave del proceso de planificación académica de horarios, desde la visión y experiencia directa de la autoridad responsable.

3.6.3 Estructura del instrumento de recolección de datos aplicados

3.6.3.1 Cuestionario para docentes

El instrumento de recolección de datos utilizado en esta investigación es un cuestionario dirigido a los docentes de la carrera de Ingeniería en Software de la ULEAM, extensión El Carmen. El cuestionario está estructurado en doce (12) preguntas cerradas, este cuestionario tiene como objetivo principal recolectar información relevante sobre el proceso actual de planificación académica de horarios, identificando sus limitaciones y la disposición del personal docente hacia la implementación de una aplicación web que optimice dicho proceso.

3.6.3.2 Entrevista dirigida al coordinador de la carrera

En esta entrevista, se abordaron 10 preguntas claves orientadas a comprender el modelo actual de planificación, identificar los principales desafíos y obtener insumos prácticos que guíen el diseño e implementación de una aplicación web que contribuya a mejorar dicho proceso.

3.6.4 Plan de recolección de datos

La recolección de datos se ejecutó de forma secuencial y organizada, en primera instancia se elaboró un cuestionario estructurado con preguntas cerradas dirigido a los docentes de la carrera de Ingeniería en Software, dicho instrumento fue implementado mediante la plataforma Google Forms, generándose un enlace de acceso que fue compartido con los participantes para su respuesta en línea durante la jornada académica, los docentes respondieron el cuestionario de manera autónoma dentro del día establecido para la recolección de la información.

Posteriormente, se aplicó una entrevista estructurada al coordinador académico de forma presencial, utilizando un documento previamente elaborado con diez preguntas orientadas a conocer las dificultades del proceso actual de planificación de horarios y la viabilidad de implementación del sistema propuesto. La entrevista tuvo una duración aproximada de 10 a 20 minutos y la información fue registrada por escrito para su posterior análisis.

Durante todo el proceso se garantizó la confidencialidad de los participantes y el uso estrictamente académico de los datos recopilados.

3.6.5 Análisis y presentación de resultados

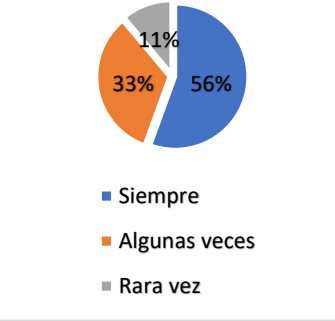
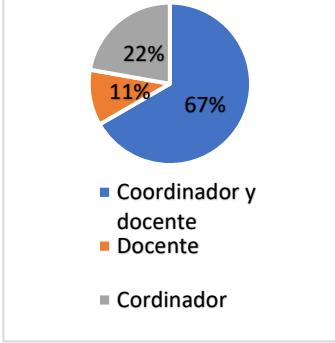
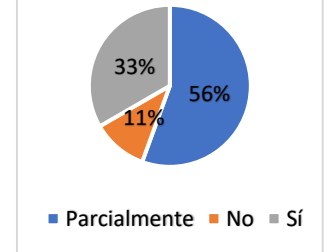
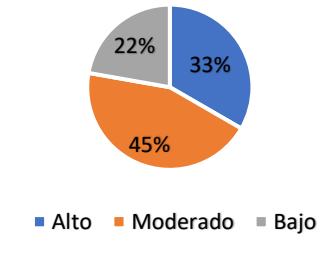
El análisis de datos permite recolectar información útil mediante datos y procedimientos estadísticos cuantitativos que permiten comunicar los hallazgos, así poder comprender todos los objetivos de la investigación.

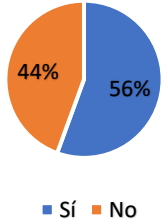
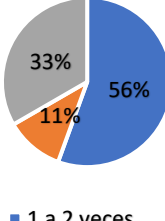
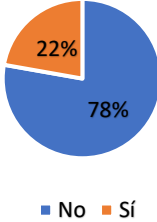
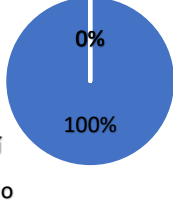
3.6.5.1 Tabulación y análisis de los datos

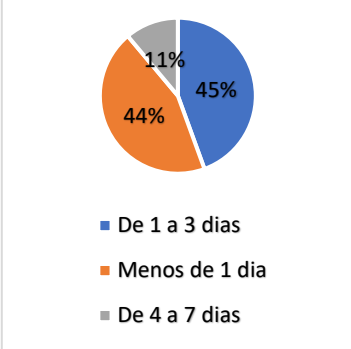
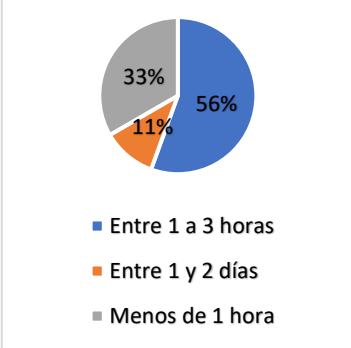

En los siguientes gráficos, se identifican los resultados de la encuesta realizada a los docentes de jornada completa de la carrera de Ingeniería en Software de la Uleam. Ext. El Carmen.

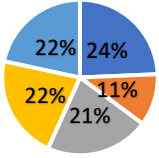
3.6.5.2 Encuesta

Tabla 1: Respuestas de la encuesta

PREGUNTA	GRÁFICO	INTERPRETACIÓN
<p>¿Con qué frecuencia participa usted en la elaboración de horarios académicos?</p>	 <p>■ Siempre ■ Algunas veces ■ Rara vez</p>	<p>La mayoría de los docentes siempre participan en la elaboración de los horarios académicos.</p>
<p>¿Quiénes realizan los horarios?</p>	 <p>■ Coordinador y docente ■ Docente ■ Cordinador</p>	<p>La mayoría indica que los horarios son realizados entre el docente y el coordinador de la carrera.</p>
<p>¿Considera que el proceso actual de elaboración de horarios es eficiente?</p>	 <p>■ Parcialmente ■ No ■ Sí</p>	<p>Según el resultado de esta pregunta la mayoría de docentes lo señala como Parcialmente eficiente.</p>
<p>¿Qué nivel de dificultad considera que tiene actualmente el proceso de planificación académica de horarios?</p>	 <p>■ Alto ■ Moderado ■ Bajo</p>	<p>Según el análisis de esta pregunta la dificultad que presenta el proceso de planificación de es Moderado, no tiene muchas complicaciones.</p>

PREGUNTA	GRÁFICO	INTERPRETACIÓN
¿Ha experimentado conflictos de horarios (choques de materias, aulas, u horarios repetidos)?	 <p>A pie chart with two segments: a blue segment representing 'Sí' at 56% and an orange segment representing 'No' at 44%. A legend below the chart identifies the colors: blue for 'Sí' and orange for 'No'.</p>	Con el resultado obtenido en esta pregunta se confirma que si existen los conflictos mencionados.
¿Con qué frecuencia los horarios planificados son modificados durante el semestre?	 <p>A pie chart with three segments: a blue segment for '1 a 2 veces' at 56%, a grey segment for 'Nunca' at 33%, and an orange segment for 'Más de 2 veces' at 11%. A legend below the chart identifies the colors: blue for '1 a 2 veces', orange for 'Más de 2 veces', and grey for 'Nunca'.</p>	Con el resultado obtenido de la sexta pregunta, se puede verificar que existen modificaciones en los horarios de 1 a 2 veces en el semestre.
¿Cuenta con acceso a un sistema o plataforma institucional para visualizar su horario?	 <p>A pie chart with two segments: a blue segment for 'No' at 78% and an orange segment for 'Sí' at 22%. A legend below the chart identifies the colors: blue for 'No' and orange for 'Sí'.</p>	Con el resultado obtenido en la séptima pregunta, se puede analizar que los docentes no tienen un acceso rápido para ver sus horarios laborables
¿Está de acuerdo con que una herramienta tecnológica podría facilitar el proceso de planificación y organización de los horarios académicos de clases en la carrera de Ingeniería en Software?	 <p>A pie chart with three segments: a blue segment for 'Sí' at 100%, a grey segment for 'No estoy de acuerdo' at 0%, and an orange segment for 'No' at 0%. A legend below the chart identifies the colors: blue for 'Sí', orange for 'No', and grey for 'No estoy de acuerdo'.</p>	Según los resultados obtenidos en esta pregunta, podemos verificar que si existe la necesidad de una herramienta tecnológica, sería una excelente idea para agilizar y organizar los horarios.

PREGUNTA	GRÁFICO	INTERPRETACIÓN
<p>¿Qué tiempo promedio se tardan actualmente en la elaboración de horarios?</p>	 <p>■ De 1 a 3 días ■ Menos de 1 día ■ De 4 a 7 días</p>	<p>Con los resultados obtenidos en la novena pregunta se puede verificar lo pacioso que es elaborar los horarios en la actualidad quitando tiempo a quien lo elabora de 1 a 3 días.</p>
<p>¿Qué tiempo promedio se tardan en realizar un cambio por choques o disponibilidad de horas al momento de la elaboración de los horarios?</p>	 <p>■ Entre 1 a 3 horas ■ Entre 1 y 2 días ■ Menos de 1 hora</p>	<p>Según el resultado obtenido en esta pregunta, podemos ver que es necesario un tiempo entre 1 a 3 horas para reajustar cambios en los horarios ya realizados.</p>
<p>¿Estaría dispuesto a utilizar una aplicación web para elaborar y consultar los horarios asignados?</p>	 <p>■ Sí ■ No ■ Tal vez</p>	<p>Según los resultados obtenidos en esta pregunta, comprendemos que si existe el interés de obtener una aplicación web de parte del docente para poder estar al tanto de sus horas asignadas.</p>

PREGUNTA	GRÁFICO	INTERPRETACIÓN
<p>¿Qué funcionalidades considera más útiles en una aplicación para planificación de horarios? (Seleccione 2 o más)</p>	 <p>A pie chart with five segments representing different functionalities. The segments are: 'Visualización del horario' (24%), 'Generación de reportes' (11%), 'Consulta de aulas asignadas' (21%), 'Acceso desde celular' (22%), and 'Notificaciones de cambio' (22%).</p> <ul style="list-style-type: none"> ■ Visualización del horario ■ Generación de reportes ■ Consulta de aulas asignadas ■ Acceso desde celular ■ Notificaciones de cambio 	<p>Según los resultados obtenidos en la pregunta actual, se puede entender que todas las opciones detalladas son muy necesarias.</p>

3.6.5.3 Entrevista

A continuación, se muestra las respuestas de la entrevista dirigida al coordinador de la carrera de Ingeniería en Software de la Uleam. Ext. El Carmen

Tabla 2: Entrevista dirigida al coordinador de la carrera de software

PREGUNTAS	RESPUESTA	INTERPRETACIÓN
<p>¿Cómo describe el proceso actual de planificación académica de horarios en la carrera de Ingeniería en Software?</p>	<p>Este proceso se realiza de forma colaborativa en reunión con todos los docentes de la carrera para la realización de este, normalmente se lo hace en una mañana o en una tarde, siendo una tarea un tanto extensa.</p>	<p>El horario es un proceso colaborativo y consensuado puesto que requiere la participación de todos los docentes de la carrera.</p>
<p>¿Qué expectativas tiene sobre el impacto de una aplicación web en el contexto académico actual de la carrera?</p>	<p>Las aplicaciones de todo tipo tanto web u otras tienen un impacto muy importante dentro de nuestro ámbito educativo y también a nivel de sociedad, obviamente tengo expectativas altas en el uso de aplicaciones.</p>	<p>Se muestran altas expectativas sobre la incorporación de las aplicaciones web en la universidad.</p>

PREGUNTAS	RESPUESTA	INTERPRETACIÓN
¿Considera que el proceso actual de elaboración de horarios es eficiente?	Totalmente eficiente no lo es, debido a que se han presentado errores en alguna ocasión que se han tenido que corregir posteriormente y esto ha causado retraso en la creación de los horarios	El proceso actual para la elaboración de horarios no es totalmente eficiente. Se señala que la presencia de errores retrasa la elaboración de los horarios y toma mucho tiempo en ser corregidos.
¿Qué nivel de dificultad considera que tiene actualmente el proceso de planificación académica de horarios?	Tiene un nivel de dificultad media el formato que utilizamos para la realización de horarios, debido a que se lo realiza en aplicación de Excel.	El proceso actual en la elaboración de los horarios tiene un nivel de dificultad media.
¿Ha experimentado conflictos de horarios (choques de materias, aulas, u horarios repetidos)?	Si he experimentado este tipo de situaciones en donde hay choques de horarios, esto debido a que al momento de diseñarlo al ser un proceso manual se han omitido algunas condiciones	Como resultado de la pregunta cinco de la entrevista, se puede claramente dar a entender los conflictos que se presentan en los horarios.
¿Con qué frecuencia los horarios planificados son modificados durante el semestre?	Durante el semestre normalmente no hay ajustes de horario o cambios de horario, pero sí lo hay al momento de crearlos hasta el momento de subirlos al sistema de gestión académica, en ese lapso si han existido modificaciones.	Se da entender que una vez creado los horarios se puede esperar sugerencias o modificaciones en un lapso de tiempo antes de ser subido al sistema donde modificarlo sería una tarea muy compleja.

PREGUNTAS	RESPUESTA	INTERPRETACIÓN
¿Considera necesario implementar una aplicación web para este proceso? ¿Por qué?	Sí considero que sería importante una aplicación de este tipo (web) al ser accesible, al ser responsiva y que puedan manejarse este tipo de actividad.	Como resultado de la pregunta siete de la entrevista, la necesidad de una aplicación web para el proceso de horarios es vista como algo importante.
¿Qué funcionalidades considera esenciales que debería tener una herramienta informática para este propósito?	Debe ser una aplicación con validaciones para evitar choques de horarios, ingresos de datos incorrectos y con bastante retroalimentación al usuario que maneja la aplicación.	Como resultado de la pregunta ocho de la entrevista, las funcionalidades esenciales para una nueva herramienta se enfocan en prevenir problemas y guiar al usuario.
¿Qué beneficios considera que podría aportar una aplicación web al proceso de planificación académica?	Los beneficios serían varios por ser una herramienta accesible desde un navegador, pudiendo obtener información en cualquier tiempo y lugar, y teniendo datos siempre a mano.	Como resultado de la pregunta nueve de la entrevista, los beneficios de una aplicación web para la planificación de horarios se enfocan en la comodidad y la facilidad de uso.
¿Qué aspectos considera importantes para lograr que el personal docente y administrativo adopte y utilice la aplicación?	Considero que uno de los fundamentos o pilares importantes para que los usuarios utilicen la aplicación es que esté bien diseñada, que realice su trabajo de una forma amigable, óptima, fácil de utilizar y que cumpla con su objetivo fundamental.	Como resultado de la pregunta diez de la entrevista, el éxito de la aplicación depende de que sea fácil de usar, para que el personal la adopte, la herramienta debe ser intuitiva, cumplir su objetivo principal de forma ágil y estar bien diseñada, haciendo que el trabajo sea más simple.

3.6.5.4 Presentación y descripción de los resultados obtenidos

Los resultados de la encuesta a docentes y la entrevista al coordinador revelan que el proceso manual actual de elaboración de horarios es ineficiente y genera conflictos como choques de materias y problemas de aulas, aumentando la carga administrativa y retrasando la planificación. Docentes muestran alta disposición para usar una aplicación web que facilite la consulta y notificación en tiempo real, mientras que el coordinador destaca la necesidad de validaciones automáticas para evitar errores. En conjunto, los resultados reflejan expectativas positivas hacia una solución tecnológica que mejore la organización, reduzca errores y optimice el tiempo en la planificación académica.

3.6.5.5 Informe final del análisis de los datos

El proceso manual de elaboración de horarios en Ingeniería en Software presenta errores y retrasos, con docentes que enfrentan conflictos y modificaciones frecuentes. Los docentes muestran disposición para usar una aplicación web que facilite la consulta y notificación de cambios.

El coordinador también reconoce la necesidad de una herramienta tecnológica que valide automáticamente los horarios, sea fácil de usar y accesible. Ambos coinciden en que esta aplicación optimizaría la organización y modernizaría los procesos administrativos universitarios.

CAPÍTULO IV

MARCO PROPOSITIVO

4.1 Introducción

Este capítulo presenta la propuesta de solución basada en la investigación sobre las dificultades en la elaboración manual de horarios académicos en la carrera de Ingeniería en Software de la ULEA, El Carmen, se propuso el desarrollo de un sistema que automatice y agilice el proceso de los horarios mejorando la eficiencia, la precisión y facilitando el uso correctivo entonces la propuesta se estructura en un modelo de desarrollo en cascada, implementando recursos y etapas para asegurar su viabilidad técnica, humana y económica, así la herramienta busca optimizar el tiempo y la planificación de horarios centralizando la información y facilitando el acceso de manera segura beneficiando tanto al coordinador como a los docentes con un sistema más intuitivo, ordenado y confiable al momento de una planificación de horarios.

4.2 Descripción de la propuesta

La propuesta consiste en desarrollar una aplicación web para automatizar la elaboración de horarios académicos en la carrera de Ingeniería en Software, reemplazando el método manual actual que es lento y propenso a errores, entonces la herramienta contará con una interfaz intuitiva para que el coordinador ingrese datos de asignaturas, docentes, aulas y paralelos generando horarios automáticos y evitando conflictos, también permitirá a los docentes ver sus horarios y recibir notificaciones de cambios en su interfaz mejorando la organización y transparencia en la creación de los horarios.

- **Asignación automática de horarios:** El sistema verificará su validez y que no existan conflictos de aulas entre paralelos o docentes al momento de generar el horario o antes de publicarlo.
- **Visualización en línea:** Los docentes podrán consultar sus horarios y también su jornada laboral completa desde cualquier dispositivo inteligente con conexión a internet.
- **Gestión centralizada en la información:** La base de datos permitirá almacenar todos los datos y administrar todos los registros de materias, docentes y aulas sin perder la información.

- **Reportes y notificaciones:** Se podrán generar reportes en formato digital además de incluir alertas en caso de modificaciones realizadas por el coordinador.
- **Facilidad de uso:** La interfaz gráfica será muy intuitiva de modo que los usuarios no requieran conocimientos técnicos avanzados para su uso.

4.3 Determinación de recursos

4.3.1 Humanos

Para el desarrollo de la propuesta fue necesario contar con la participación de distintos actores, cada uno con un rol específico dentro del proceso. En este caso, se identifican los siguientes:

Coordinador de la carrera: responsable de proporcionar información sobre el proceso actual de elaboración de horarios y validar los requisitos de la aplicación web.

Docentes de jornada completa: participaron en la encuesta, aportando información clave para identificar necesidades y validar las funcionalidades requeridas en el sistema.

Tutora del proyecto: encargada de la supervisión académica y de la revisión del documento de investigación.

Desarrollador (tesista): responsable de la programación, pruebas y puesta en marcha de la aplicación web.

4.3.2 Tecnológicos

El desarrollo de la aplicación web para la elaboración de horarios requiere de un conjunto de recursos tecnológicos que permitan llevar a cabo tanto el proceso de programación como las pruebas y la futura implementación. Estos recursos se dividen en hardware, software y plataformas de apoyo, los cuales son indispensables para asegurar el correcto funcionamiento del proyecto.

4.3.2.1 Recursos de hardware

Para el desarrollo y las pruebas de funcionamiento de la aplicación web se utilizó un único equipo de cómputo personal, el cual permitió la programación, ejecución local del sistema y verificación de su comportamiento. El equipo corresponde a un computador portátil con procesador AMD Ryzen 7 7730U de 64 bits a 2.00 GHz y 16 GB de memoria RAM, características suficientes para la ejecución del entorno de desarrollo y la base de datos en pruebas locales.

Adicionalmente, se empleó un dispositivo móvil personal para realizar pruebas de acceso remoto mediante conexión a internet, permitiendo verificar la correcta visualización y funcionamiento del sistema desde diferentes resoluciones de pantalla y condiciones de conectividad.

El acceso al sistema se realizó a través de conexión a red e internet, lo cual permitió comprobar el funcionamiento tanto en entorno local como en entorno de alojamiento web.

4.3.2.2 Recursos de software

- a) **Lenguajes de programación:** HTML5, CSS3 y JavaScript para el desarrollo del frontend; PHP para el backend.
- b) **Framework de diseño:** Bootstrap para la estructuración y adaptación responsiva de la interfaz gráfica.
- c) **Gestor de base de datos:** MySQL para el almacenamiento de la información correspondiente a docentes, materias, aulas y horarios académicos.
- d) **Entorno de desarrollo:** Visual Studio Code como editor de código fuente.
- e) **Servidor local:** XAMPP para la ejecución del servidor Apache y la base de datos en pruebas locales.
- f) **Navegador web:** Firefox para la verificación del funcionamiento del sistema durante el desarrollo.

4.3.3 Económicos

El desarrollo de la propuesta también implica considerar los recursos económicos necesarios para su ejecución. Si bien este proyecto se realiza en el marco académico y no requiere una gran inversión en infraestructura, es importante estimar los costos que se generarían en caso de implementarlo de forma real. Para ello, se han tomado en cuenta tres aspectos principales: infraestructura tecnológica, servicios de software y horas de trabajo del tesista.

Tabla 3: Recursos económicos

Concepto	Cantidad	Costo unt	Costo total	Observación
Se requiere un computador personal para desarrollo,	1	800,00	800,00	Equipo disponible del tesista

Concepto	Cantidad	Costo unt	Costo total	Observación
Conexión a internet	6 meses	24,00	144,00	Servicio necesario durante el desarrollo
Electricidad e insumos	6 meses	15,00	90,00	Consumo adicional en jornadas de desarrollo
Horas de trabajo del tesista	384 h	3.00	1.152	Valor estimado del trabajo profesional
Total, estimado			2.186	

4.4 Etapas de acción para el desarrollo de la propuesta

La aplicación web para la elaboración de horarios se desarrollará mediante el modelo en cascada, siguiendo un proceso secuencial y estructurado. Este enfoque, adecuado para un proyecto individual, permite avanzar ordenadamente, integrando los requerimientos obtenidos en la investigación de campo mediante encuestas y entrevistas.

4.4.1 Fase I: Requisitos

4.4.1.1 Recopilación de Requisitos

Durante esta etapa se llevaron a cabo diversas actividades para identificar las necesidades, limitaciones y expectativas en la elaboración de horarios académicos en la carrera de Ingeniería de Software de la ULEAM, Extensión El Carmen. Se emplearon diferentes herramientas para recolectar información crucial, garantizando que la propuesta de la aplicación web se fundamentara en exigencias auténticas y confirmadas por los que toman parte directa del proceso académico.

Se llevó a cabo una entrevista con el coordinador de la carrera para determinar los retos más importantes de hoy en día y establecer rasgos esenciales para la herramienta digital, también se realizó una encuesta a los docentes que trabajan todo el día, lo que permitió obtener información cuantitativa y cualitativa que demostró la necesidad de un sistema eficaz en rendimiento para comprobar y validar horarios, reduciendo errores, el estudio de documentos institucionales, como las mallas curriculares y los horarios oficiales, posibilitó definir los parámetros básicos que debe tomar en cuenta el sistema, tales como disponibilidad de aulas, asignaturas y maestros.

Con base en esta información, se elaboraron historias de usuario que reflejan los anhelos del coordinador y de los maestros, lo cual ayuda a señalar los requisitos funcionales y no funcionales, una interfaz amigable, la creación automática de horarios que no choquen entre sí, la comprobación de las cargas académicas y la opción de hacer ajustes específicos sin impactar en el plan general son algunas de sus características sobresalientes, este proceso integral garantiza que la solución tecnológica responda de manera eficaz a las necesidades del contexto académico.

4.4.1.2 Análisis de Documentos

Para la propuesta, se analizó la documentación oficial de la carrera de Ingeniería en Software en la ULEAM, Ext. El Carmen, clave para comprender la estructura académica y la organización que afectan la planificación de horarios, tanto en Ingeniería en Software como en Tecnologías de la Información.

La complejidad de crear horarios, debido a la diversidad de materias y profesores, hace que se eleven las posibilidades de solapamientos y sobrecargas, la malla curricular de Ingeniería en Software, con múltiples niveles y turnos, tanto por la mañana como por la tarde, lo evidencia, a pesar de que las Tecnologías de la Información están cerrándose, su estructura parecida resalta la necesidad de una solución escalable para los dos programas, el propósito principal es crear una solución que esté ajustada a las especificidades y retos de ambas carreras, con el fin de mejorar la gestión y planificación de los horarios académicos.

4.4.1.3. Análisis de Requisitos

El análisis de requisitos constituye una etapa clave del desarrollo, permite identificar de manera precisa las funciones que debe cumplir la aplicación web y las características que aseguren su correcto funcionamiento. A continuación, se presentan los requisitos funcionales y no funcionales del sistema, elaborados en base a la información recabada en la encuesta a los docentes, la entrevista al coordinador, y la documentación institucional.

4.4.1.4. Priorización de Requisitos

La priorización de requisitos es un proceso esencial en el desarrollo de sistemas que ordena y selecciona las funcionalidades según su importancia, urgencia y viabilidad, permitiendo enfocar recursos en lo que aporta mayor valor al usuario y garantiza el éxito del proyecto. Facilita la toma de decisiones a lo largo del desarrollo, ayudando a manejar cambios, reducir costos y cumplir con los plazos.

Con el propósito de efectuar una priorización eficiente, se consideran criterios como el impacto en la satisfacción del usuario, los costos, la complejidad técnica y los criterios para el negocio, incluyen las opiniones de desarrolladores, coordinadores y docentes para que las expectativas sean acordes con la realidad, esta perspectiva de cooperación permite reducir los riesgos al mínimo y aumentar al máximo las ventajas, la priorización, en el contexto de la implementación de horarios académicos, determina las funcionalidades esenciales, por ejemplo, la creación y validación automática de conflictos y esto garantiza que los sectores esenciales estén funcionando desde etapas tempranas con el objetivo de optimizar la eficacia y la experiencia del usuario.

4.4.1.5 Priorización de requisitos funcionales

- a) Gestión de usuarios
- b) Diseño automático de horarios
- c) Botón de generación automática
- d) Diseño manual de horarios
- e) Botón de creación manual
- f) Edición de horarios
- g) Publicación de horarios
- h) Visualización de horarios
- i) Descarga de horarios

4.4.1.6 Priorización de requisitos no funcionales

- a) Usabilidad
- b) Rendimiento
- c) Seguridad
- d) Disponibilidad
- e) Escalabilidad
- f) Mantenibilidad
- g) Compatibilidad

4.4.2. Fase II: Análisis

4.4.2.1. Análisis de Sistemas:

El análisis muestra que la elaboración manual de horarios en la carrera de Ingeniería de Software de la ULEAM-Extensión El Carmen, presenta limitaciones y genera sobrecarga administrativa, afectando a coordinadores y docentes, aunque se requiere más evidencia sobre

los retrasos académicos. Como solución, se propone una aplicación web para automatizar y simplificar la creación y edición de horarios, ofreciendo además a los docentes acceso seguro para consultar sus asignaciones, aunque es necesario validar su viabilidad.

4.4.2.1.1 Casos de uso principales

4.4.2.1.3 Descripción gráfica

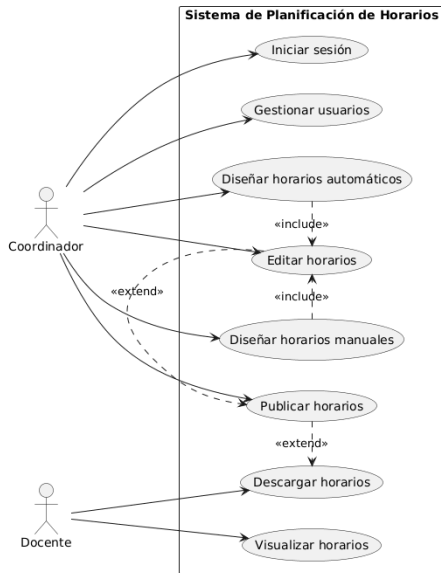


Ilustración 2: Sistema de planificación de horarios

4.4.2.1.4 Casos de uso del sistema

Caso de uso 1: Iniciar sesión

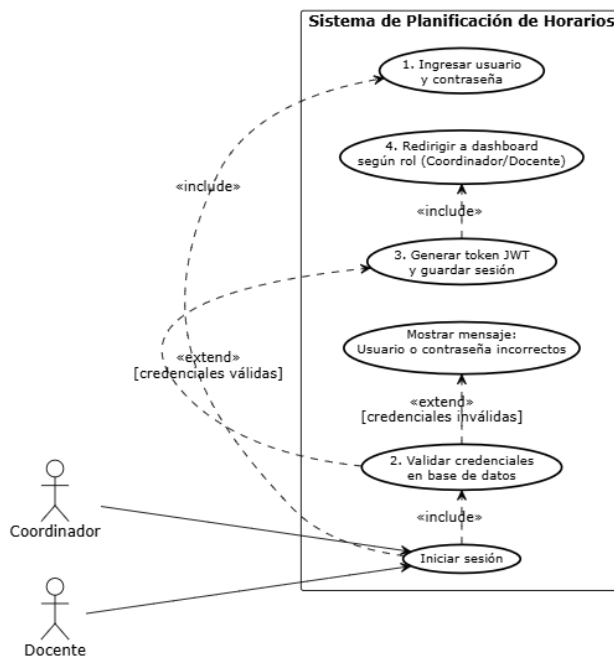


Ilustración 3: Caso de uso iniciar sesión

Caso de uso 2: Registrar información académica



Ilustración 4: Caso de uso registrar información académica

Caso de uso 3: Generar horario (automático o manual)

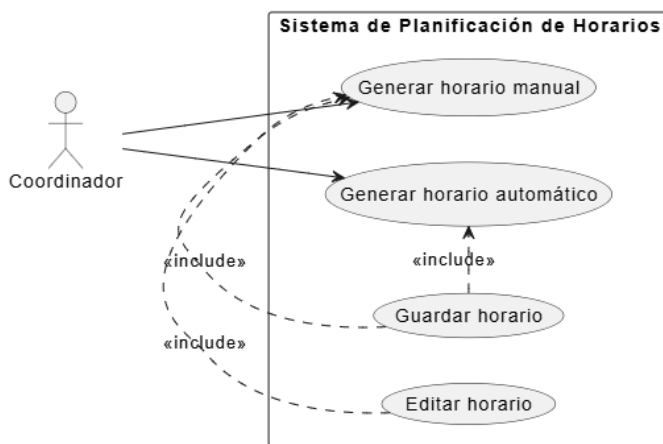


Ilustración 5: Caso de uso generar horario automático o manual

Caso de uso 4: Publicar y consultar horario

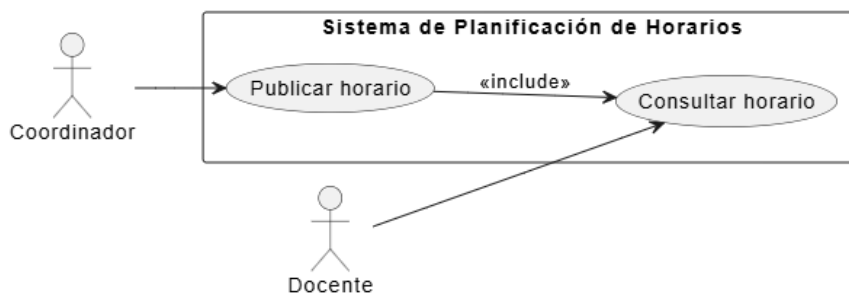


Ilustración 6: Caso de uso publicar y consultar horario

4.4.2.1.5 Diagramas de secuencia

Diagrama 1: Login de Usuario

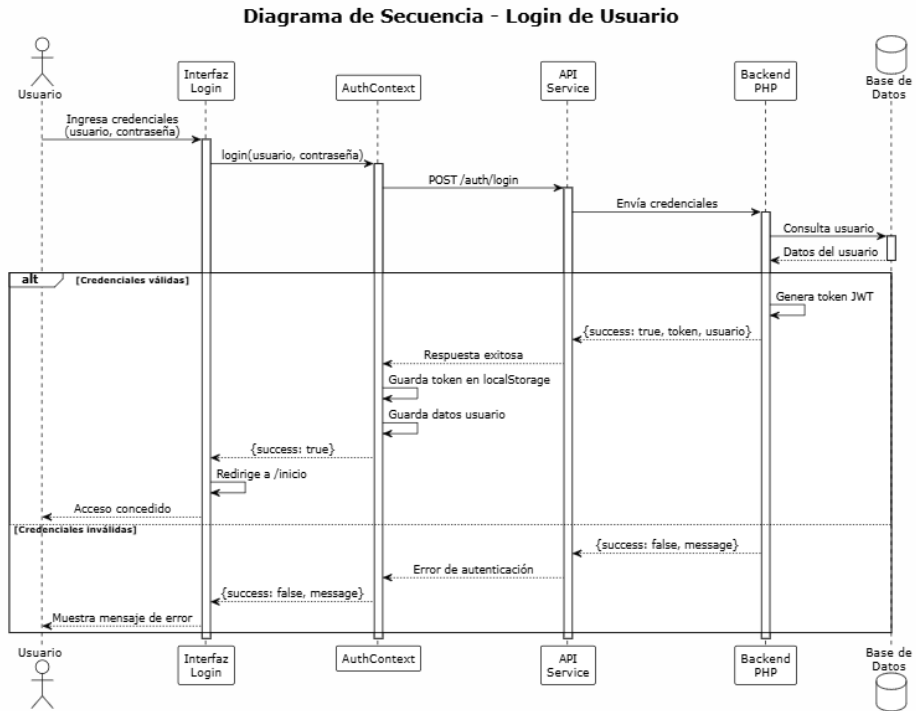


Ilustración 7: Diagrama de secuencia login de usuario

Diagrama 2: Crear Horario

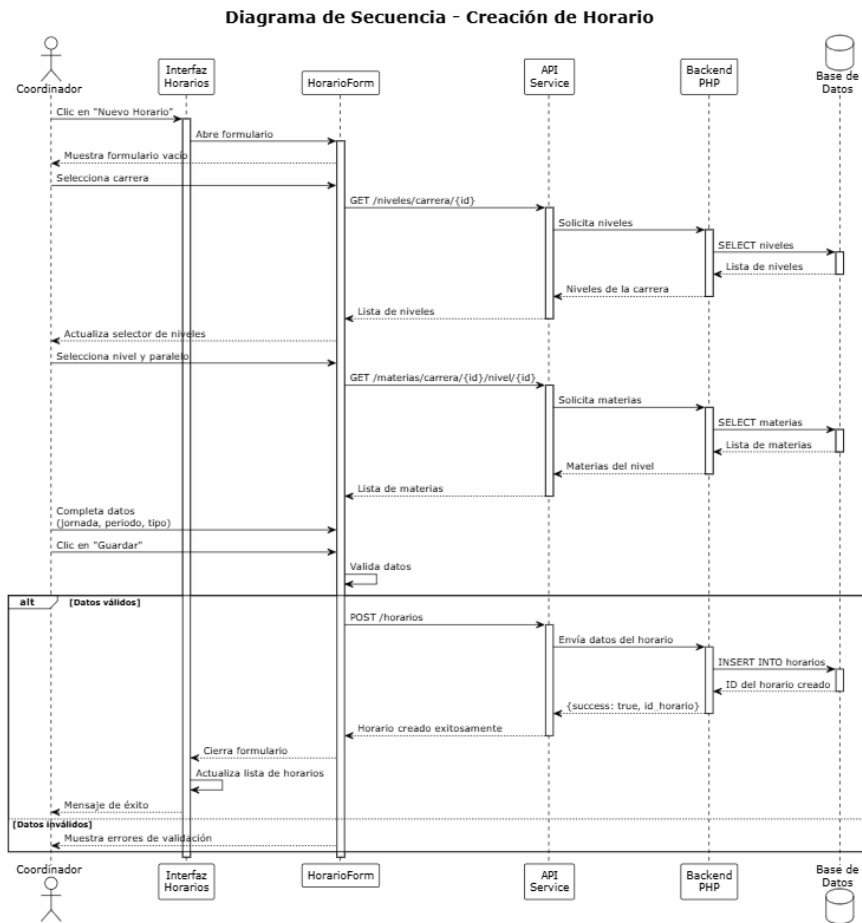


Ilustración 8: Diagrama de secuencia de crear horario

Diagrama 4: Generar Reporte PDF

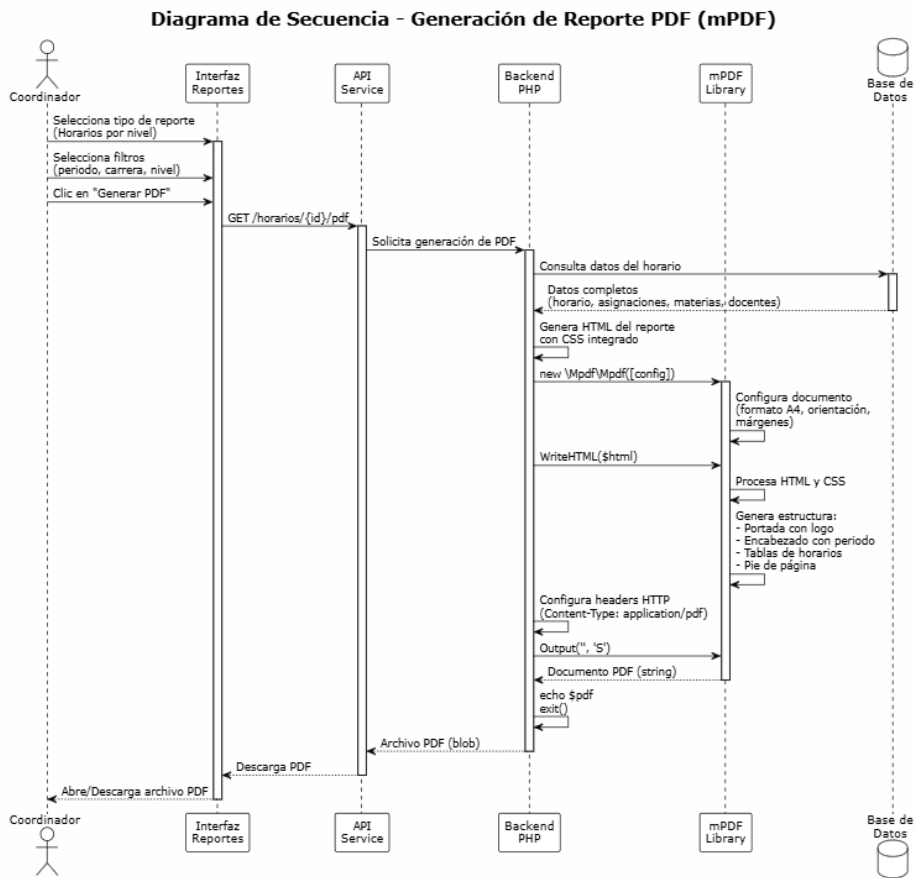


Ilustración 9: Diagrama de secuencia generar reporte

4.4.2.1.6 Diagramas de estado

Diagrama 1: Ciclo de Vida de un Horario

Diagrama de Estado - Ciclo de Vida de un Horario

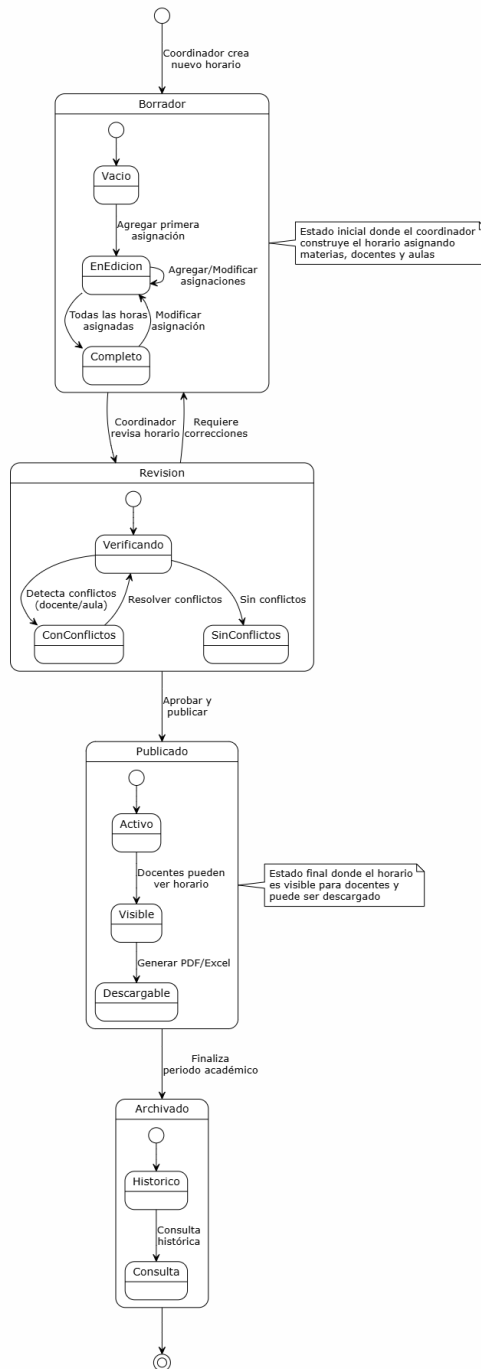


Ilustración 10: Diagrama de estado de ciclo de vida de un horario

Diagrama 2: Ciclo de Vida de un Usuario

Diagrama de Estado - Ciclo de Vida de un Usuario

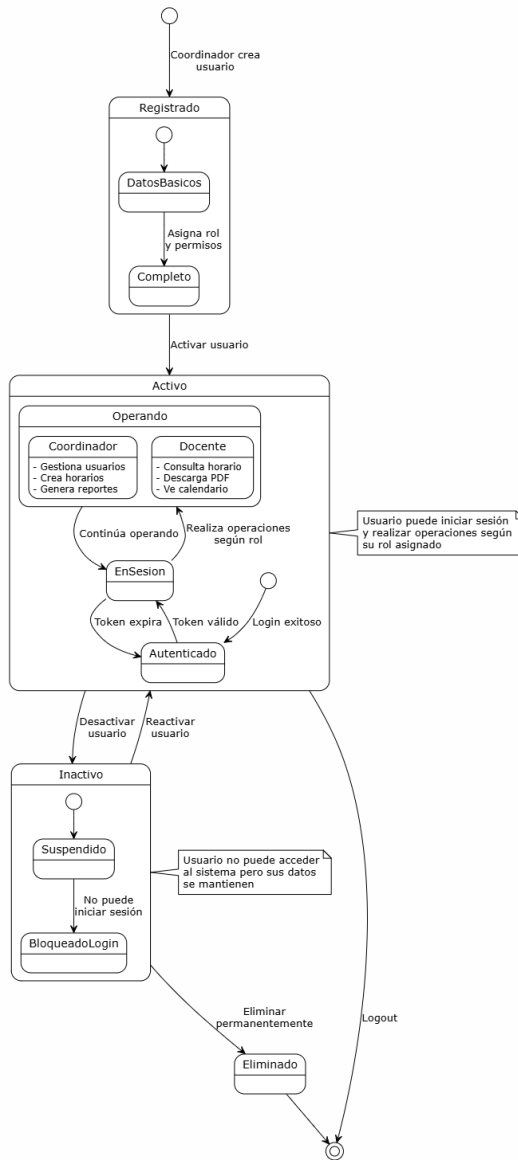


Ilustración 11: Diagrama de estado de ciclo de vida de un usuario

Diagrama 3: Estado de Asignación de Materia

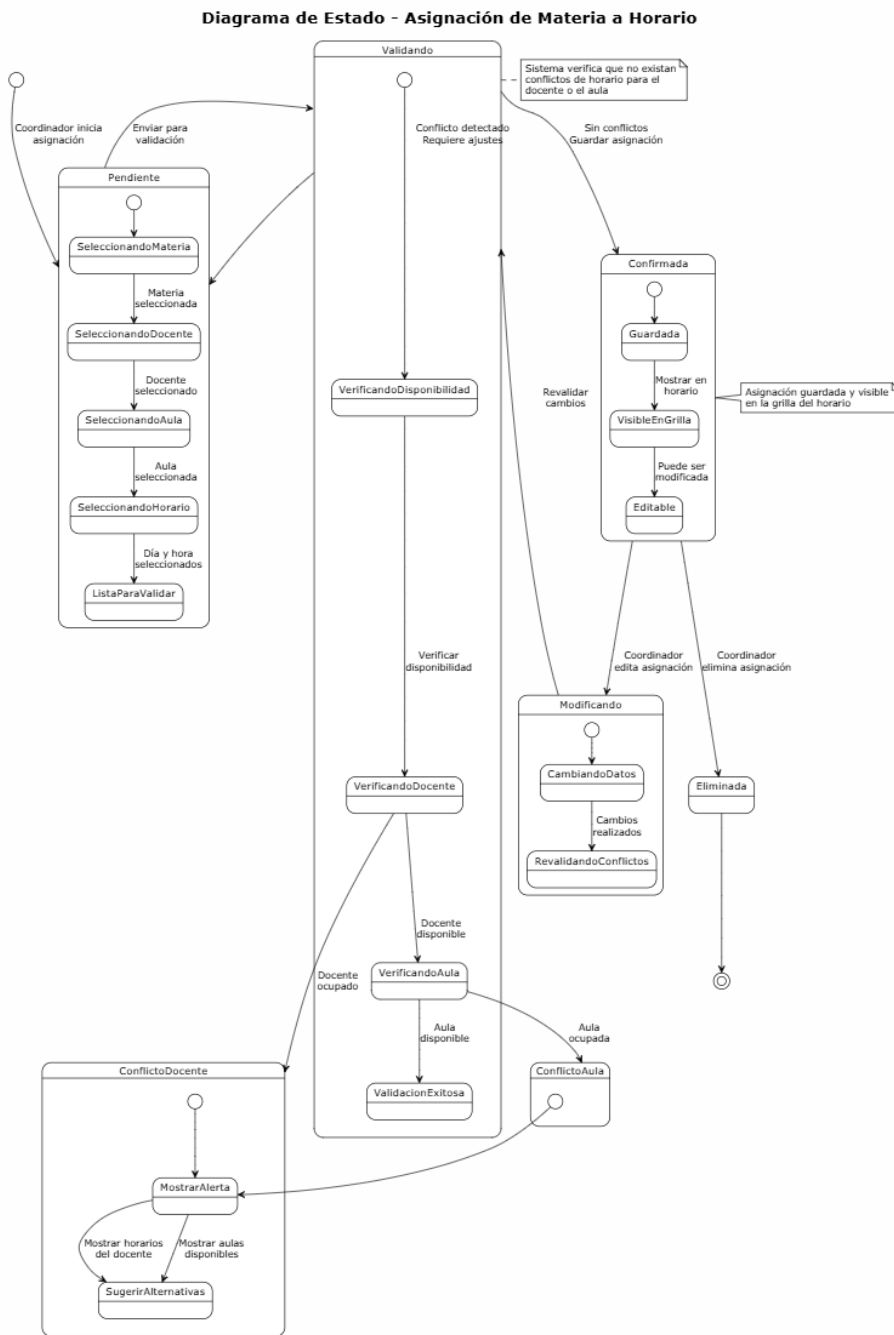


Ilustración 12: Diagrama de estado de asignación de materia a horario

Diagrama 4: Estado de Período Académico

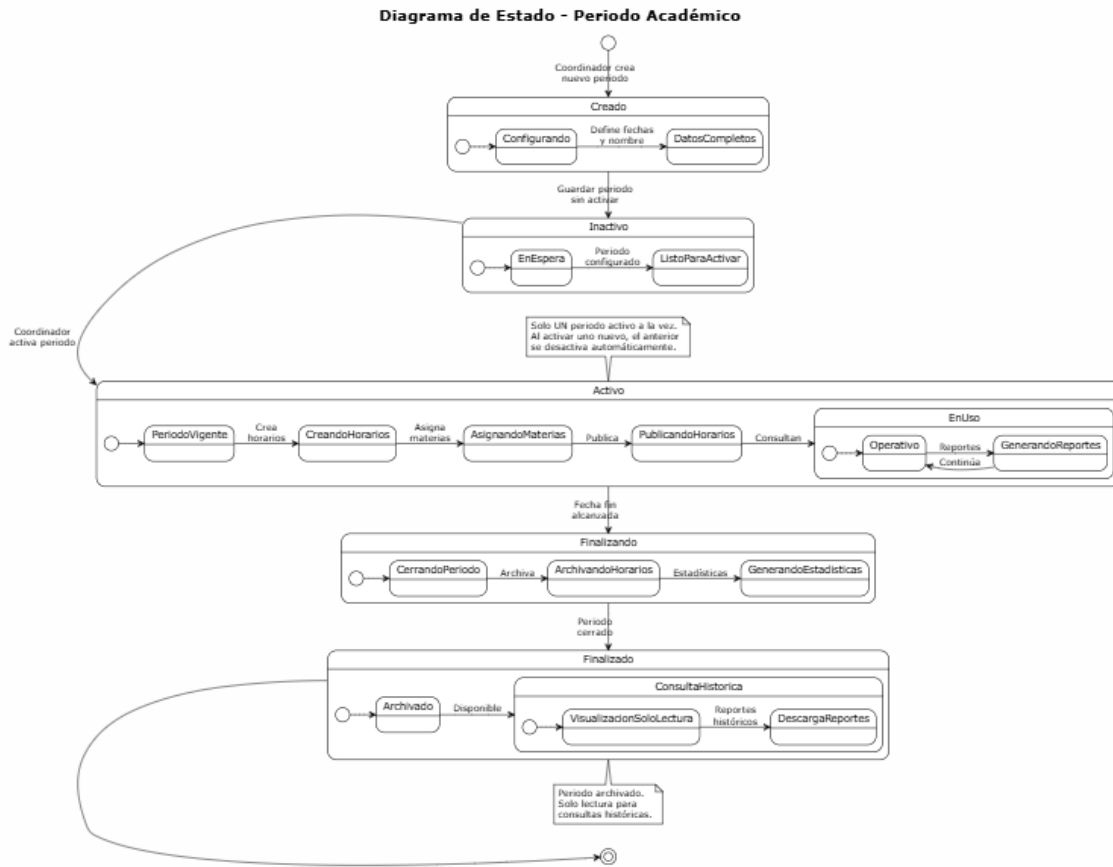


Ilustración 13: Diagrama de estado de estado de periodo académico

Diagrama 6: Detección y Resolución de Conflictos

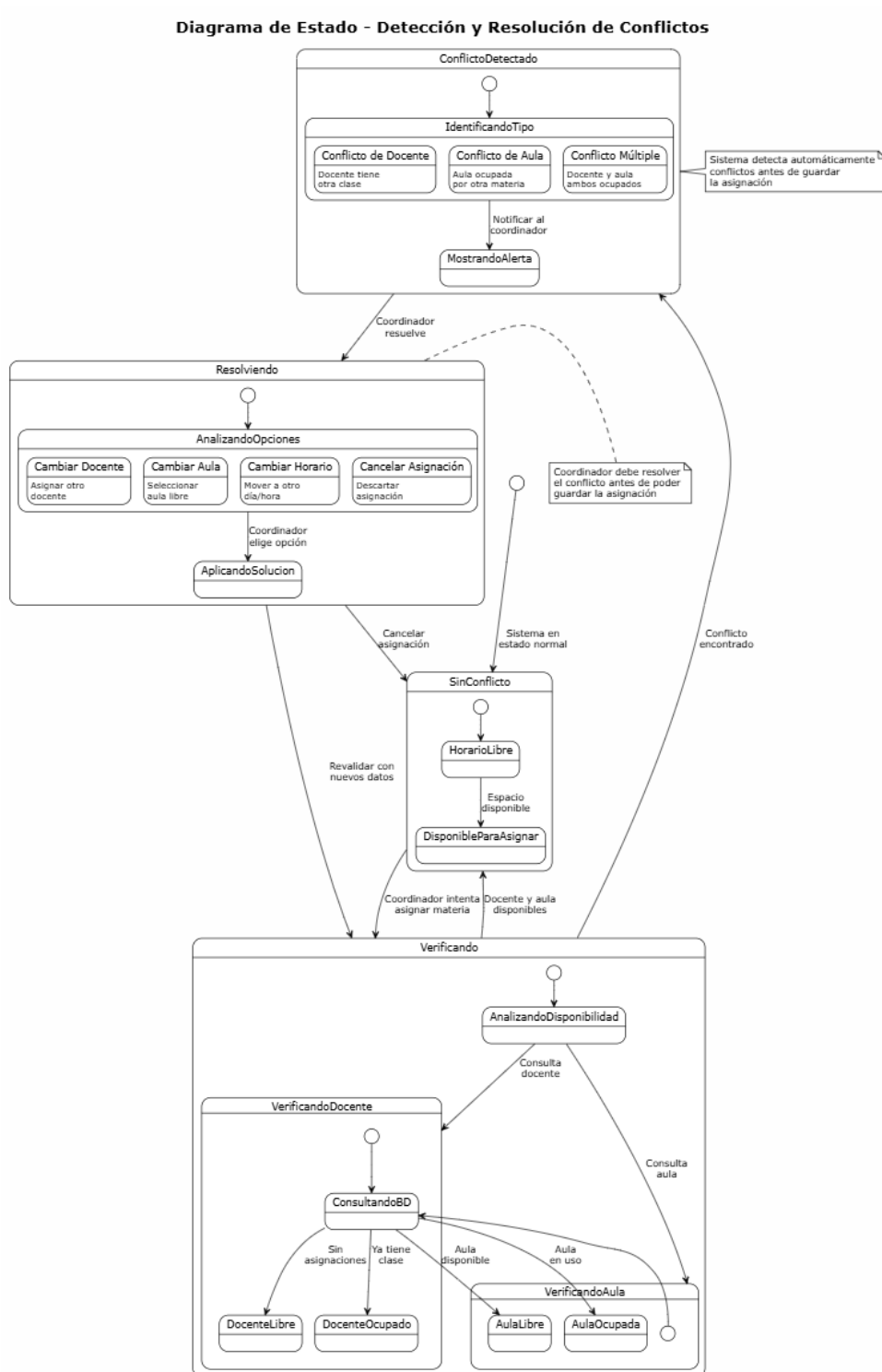


Ilustración 14: Diagrama de estado de detección y resolución de conflictos

4.4.2.2. Especificación de Requisitos

4.4.2.2.1. Requisitos Funcionales

Tabla 4: Especificación de requisitos funcionales

N.º	Requisito Funcional	Descripción
RF1	Gestión de usuarios	Acceso seguro mediante credenciales únicas para coordinador (administrador) y docentes.
RF2	Diseño automático de horarios	Permite seleccionar carrera, nivel, jornada y aula; genera automáticamente horarios asignando docentes según los docentes registrados, evitando choques de horas y aulas.
RF3	Botón de generación automática	Incluye un botón Generar automático que ejecuta el proceso de creación automática de horarios.
RF4	Diseño manual de horarios	Permite que el coordinador asigne manualmente docentes, materias y aulas con validaciones que eviten choques, las horas por materia se asignan al crear una materia.
RF5	Botón de creación manual	Incluye un botón Crear horario manual que valida y guarda la configuración ingresada.
RF6	Modificación de horarios	Capacidad para cambiar los horarios creados (ya sean manuales o automáticos) antes de ser publicados y revisados por los docentes, lo que incluye alteraciones en profesores, asignaturas, aulas y horas.
RF7	Publicación de horarios	Permite que el horario aprobado se publique y sea visible en el sistema.
RF8	Visualización de horarios	Los maestros con sus credenciales podrán ver solo sus propios horarios por nivel o carrera.
RF9	Descarga de horarios	Los horarios tanto previos como posteriores a su publicación estarán disponibles para descarga en los formatos PDF, Excel, CSV y HTML.
RF10	Registro del horario vigente	El sistema mostrará únicamente el horario publicado más reciente, evitando confusiones con versiones anteriores.

4.4.2.2. Requisitos No Funcionales

Tabla 5: Especificación de requisitos no funcionales

N.º	Requisito No Funcional	Descripción
RNF1	Usabilidad	Interfaz simple, intuitiva y de fácil manejo para coordinadores y docentes.
RNF2	Rendimiento	La generación automática de horarios debe realizarse en menos de 5 segundos.
RNF3	Seguridad	Acceso protegido con credenciales, cifrado de información sensible y control de roles.
RNF4	Disponibilidad	Acceso desde cualquier navegador moderno y dispositivo con conexión a internet.
RNF5	Escalabilidad	Capacidad de ampliar la aplicación para incluir nuevas carreras o extensiones universitarias.
RNF6	Mantenibilidad	Código modular y documentado que facilite futuras actualizaciones o mejoras.
RNF7	Compatibilidad	Los formatos de exportación (PDF, Excel, CSV y HTML) deben garantizar legibilidad en distintos entornos y dispositivos.

4.4.3. Fase III: Diseño

4.4.3.1. Diseño del Sistema:

El sistema propuesto para la gestión y elaboración de horarios académicos en la carrera de Ingeniería en Software de la ULEAM, ext. El Carmen, se estructura bajo una arquitectura cliente-servidor, permitiendo una separación clara entre la lógica de presentación, la lógica de negocio y el manejo de datos. Esta organización garantiza mayor orden, escalabilidad y mantenimiento adecuado conforme el sistema crezca o se requieran nuevas funcionalidades.

4.4.3.2. Arquitectura del Sistema

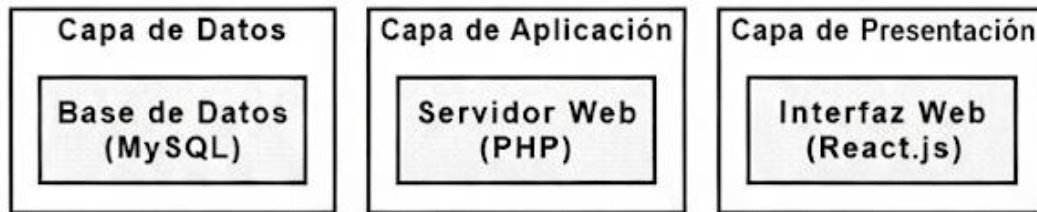


Ilustración 15: Arquitectura del sistema modelo de 3 capas

La capa de datos está constituida por una base de datos MySQL administrada desde XAMPP, donde se guarda toda la información académica y operativa del sistema. Esta base garantiza la consistencia y disponibilidad de datos cruciales como horarios, asignaciones, períodos académicos y estructura institucional.

La capa de aplicación opera como un intermediario entre el frontend y la base de datos, desarrollándose en PHP y funcionando sobre XAMPP. En esta capa se lleva a cabo la autenticación mediante tokens JWT, se valida información académica, se aplica la lógica necesaria para generar horarios automáticos sin conflictos, y se gestiona tanto a los usuarios como a los roles. Esta capa sigue el estándar REST y cuenta con una estructura organizada de controladores, modelos y servicios, lo que facilita tanto la escalabilidad como el mantenimiento del sistema.

Finalmente, la capa de presentación, construida con React.js, actúa como la interfaz directa para el usuario, gestionando entradas a través de formularios que envían solicitudes al backend. Esta capa abarca funcionalidades que van desde el inicio de sesión hasta la creación y edición de horarios, el registro de usuarios y la descarga de reportes en distintos formatos, asegurando una comunicación continua con el servidor para proporcionar una experiencia interactiva y dinámica.

4.4.3.3. Diseño de Bases de Datos

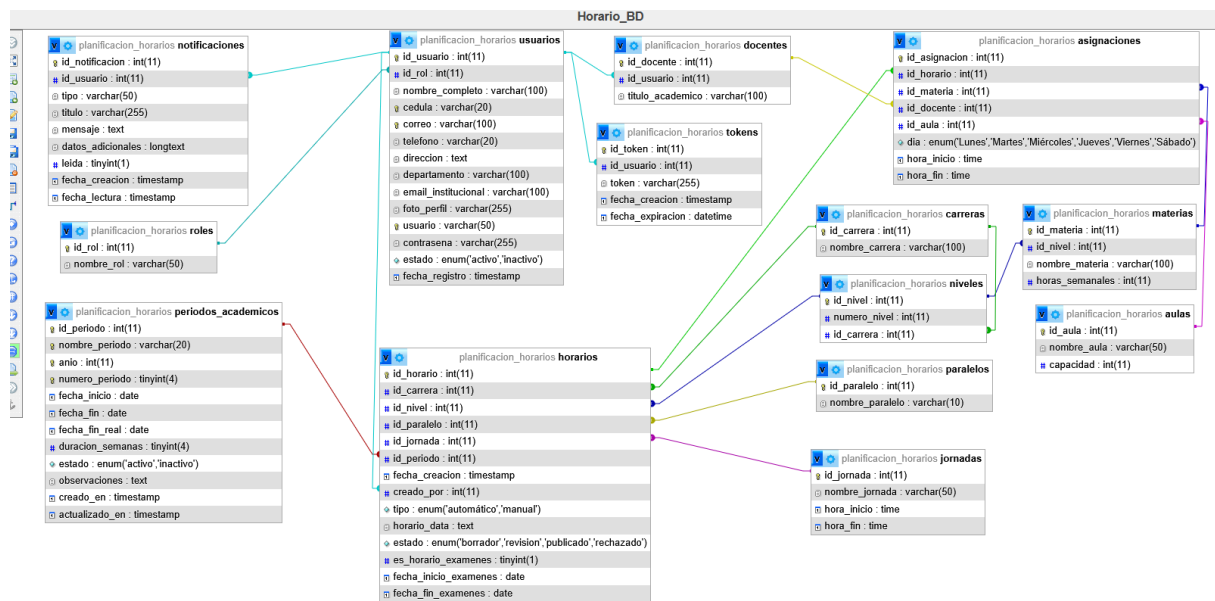


Ilustración 16: Diseño de base de datos Modelo entidad relación

4.4.3.4. Diseño de la Interfaz de Usuario

El Sistema de Planificación de Horarios presenta un diseño de interfaz de usuario fundamentado en principios de usabilidad, accesibilidad y experiencia del usuario (UX), con el objetivo de proporcionar una herramienta que sea intuitiva y eficiente para la gestión académica. Esta interfaz ha sido desarrollada utilizando React y Material-UI, lo que garantiza un diseño moderno, adaptable y consistente en todos los módulos del sistema.

4.4.3.5. Diseño de UX

a) Colores:

- **Color Rojo:** Este color se utiliza para identificar la apariencia más llamativa a la institución, se diseñó para los encabezados de los dashboards y los accesos directos de cada rol, además es el color que identifica los temas de cada módulo.
- **Color Blanco:** Este color se utiliza en el sistema para identificar los fondos de pantalla, también para facilitar la lectura de los textos y poder identificar las divisiones de los paneles.
- **Color Verde:** Este color se utiliza para indicar los estados de los usuarios como ACTIVO y para identificar el tipo de rol. El verde transmite que un elemento está habilitado o funcionando correctamente en el sistema.

- **Color de texto Negro y Gris oscuro:** Este color se utiliza para los textos los cuales, al tener una interfaz de fondo blanco, facilitan la lectura y comprensión de una óptima visibilidad al momento de la lectura.

b) Tipografía:

- **Fuente principal (Roboto):** La tipografía seleccionada es Roboto, una fuente sans-serif que se caracteriza por su modernidad, claridad y facilidad de lectura. Se utiliza de manera consistente en todos los textos de la interfaz para asegurar una apariencia profesional y uniforme.
- **Título principal:** Se establecen tamaños entre 24 y 28 px en color rojo ULEAM para resaltar claramente elementos como Gestión de Usuarios y Gestión de Horarios.
- **Encabezado de tablas:** Estos se aplican en columnas de las tablas como títulos principales para identificar los datos puestos de color blanco sobre un fondo rojo de 13 a 14 px.
- **Texto y contenido de tablas:** Se aconseja emplear un tamaño de 13-14 px en negro para garantizar una lectura clara y concisa para poder identificar el funcionamiento de cada acción.
- **Texto de botones:** Está diseñado de 13 y 14 px en blanco y con un peso medio, utilizado en los usuarios en botones identificadores como nuevo usuario y solo activos.
- **Texto de badges de estado:** Se sugiere un tamaño de 11 y 12 px en color blanco sobre fondo verde, indicando el estado como ACTIVO en mayúsculas.

4.4.3.6. Prototipos

Mapa de navegación:

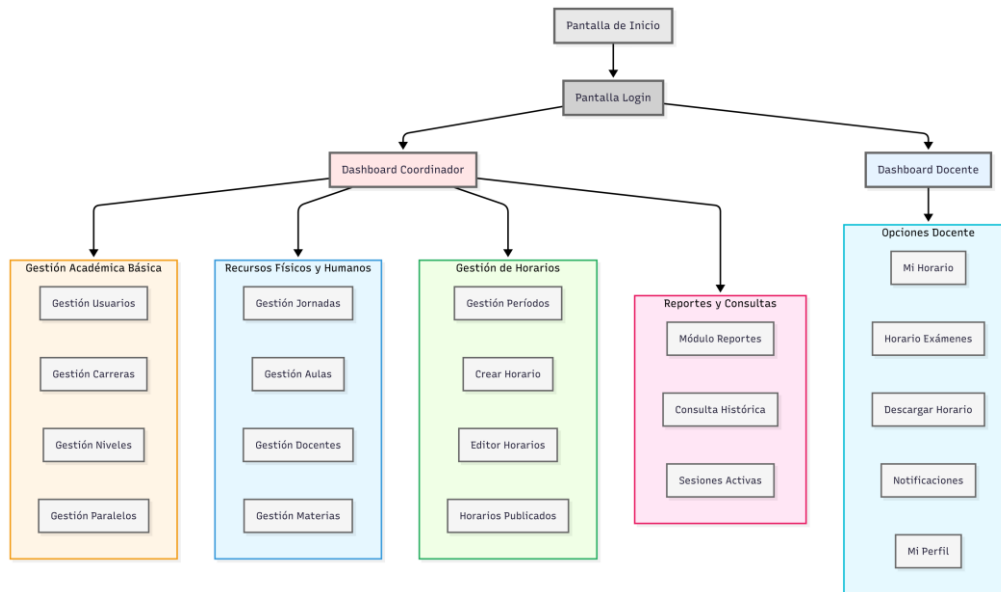


Ilustración 17: Mapa de navegación del sistema

4.4.3.7 Interfaces de entrada:

4.4.3.7.1 Pantalla de Inicio de Sesión

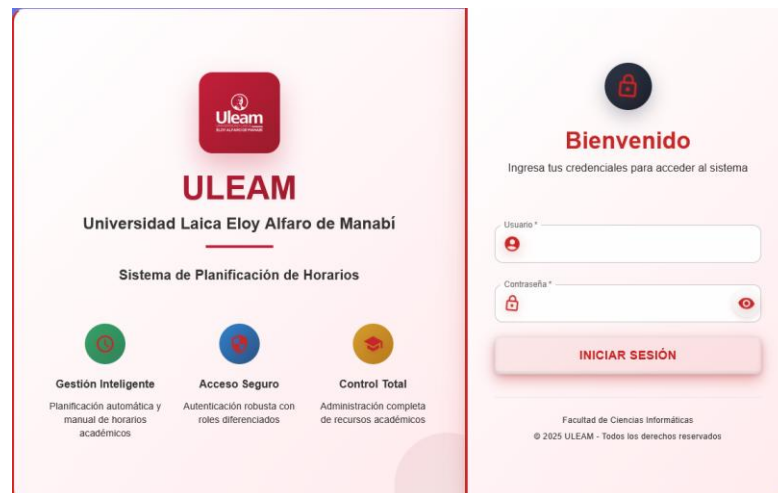


Ilustración 18: Interfaz de inicio de sesión

La pantalla de inicio de sesión representa un punto de entrada al sistema con un diseño limpio y profesional, incluye campos para usuario y contraseña, con opción de mostrar u ocultar la contraseña, un botón destacado para iniciar sesión y mensajes de error para credenciales incorrectas, técnicamente, cuenta con validación en tiempo real, feedback visual, diseño adaptable a diferentes dispositivos y seguridad mediante encriptación de contraseñas.

4.4.3.7.2 Dashboard del Coordinador

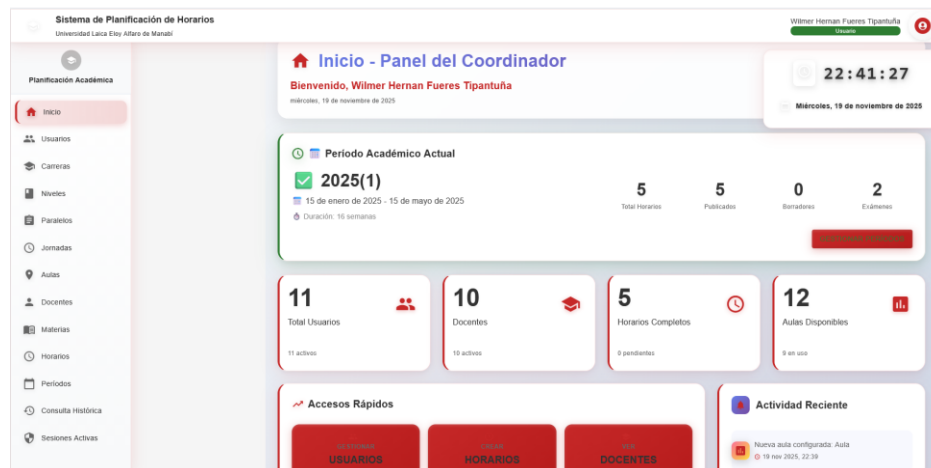


Ilustración 19: Interfaz dashboard del coordinador

El dashboard del coordinador ofrece una vista general del sistema con acceso rápido a sus funciones clave. Cuenta con una barra de navegación lateral, encabezado con información del usuario y periodo activo, tarjetas de estadísticas, widgets detallados del periodo académico, accesos rápidos, actividad reciente y un reloj en tiempo real. Técnicamente, permite actualización dinámica de datos, gráficos interactivos, navegación intuitiva y visualización clara del estado del sistema.

4.4.3.7.3 Gestión de Periodos Académicos

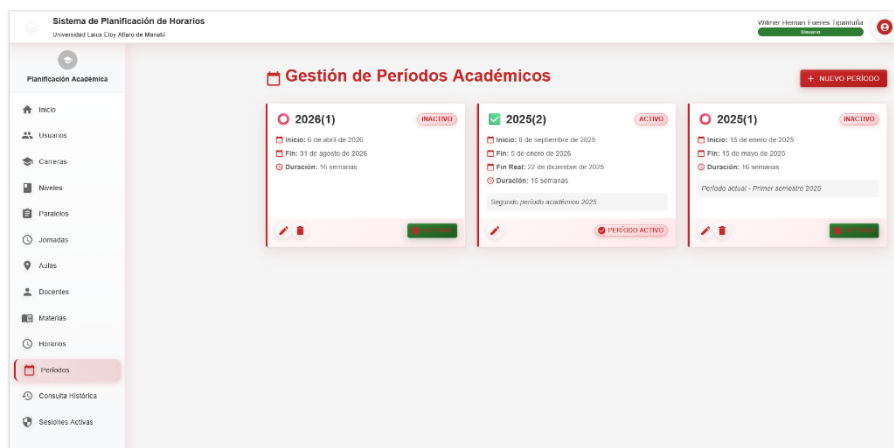
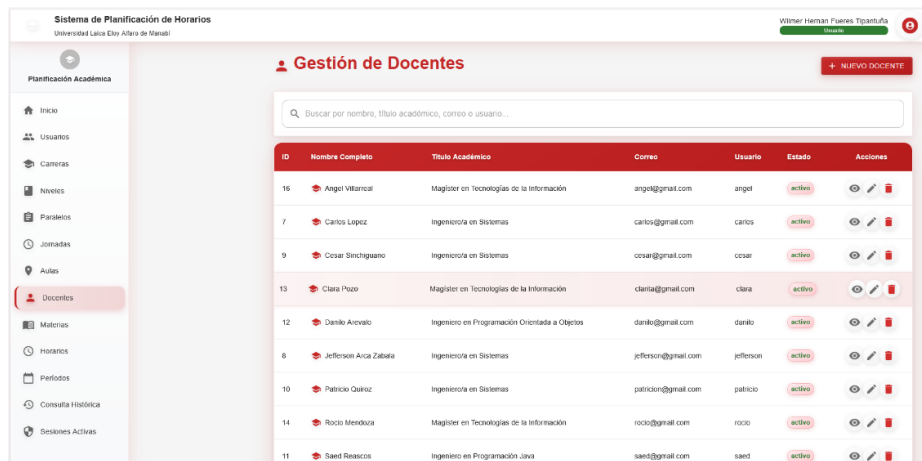


Ilustración 20: Interfaz de periodos académicos

La interfaz de gestión de periodos permite crear, activar y administrar los periodos lectivos mediante una lista visual con tarjetas que muestran detalles como nombre, fechas y estado. Incluye indicadores del periodo activo, estadísticas, botones para editar, eliminar, activar y ver horarios, además de filtros y búsqueda.

4.4.3.7.4 Gestión de Información Académica

Gestión de Docentes.



The screenshot shows the 'Gestión de Docentes' (Teacher Management) interface. It features a search bar at the top with the placeholder text 'Buscar por nombre, título académico, correo o usuario...'. Below the search bar is a table with the following columns: ID, Nombre Completo, Título Académico, Correo, Usuario, Estado, and Acciones. The table contains 11 rows of data for various teachers.

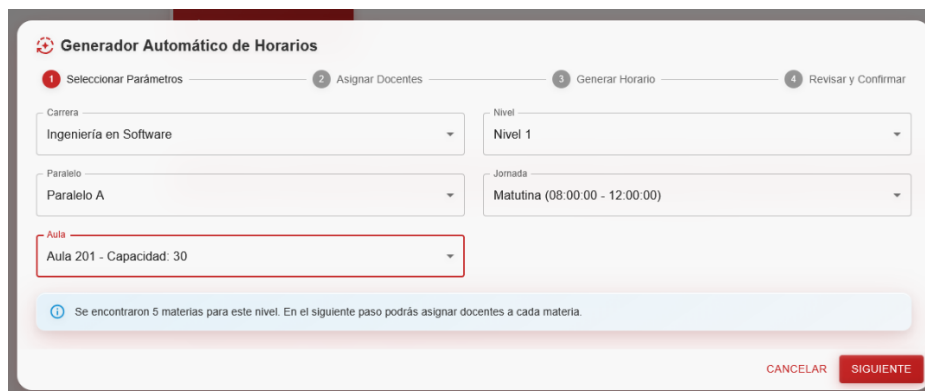
ID	Nombre Completo	Título Académico	Correo	Usuario	Estado	Acciones
16	Angel Villarreal	Magister en Tecnologías de la Información	angel@gmail.com	angel	activo	[icon]
7	Carlos Lopez	Ingeniero en Sistemas	carlos@gmail.com	carlos	activo	[icon]
9	Cesar Sinchigano	Ingeniero en Sistemas	cesar@gmail.com	cesar	activo	[icon]
13	Clara Pazo	Magister en Tecnologías de la Información	clara@gmail.com	clara	activo	[icon]
12	Danielo Arceño	Ingeniero en Programación Orientada a Objetos	danielo@gmail.com	danielo	activo	[icon]
8	Jefferson Avila Zabala	Ingeniero en Sistemas	jefferson@gmail.com	jefferson	activo	[icon]
10	Patricio Quiroz	Ingeniero en Sistemas	patricio@gmail.com	patricio	activo	[icon]
14	Rocio Menduza	Magister en Tecnologías de la Información	roco@gmail.com	roco	activo	[icon]
11	Saad Reasco	Ingeniero en Programación Java	saad@gmail.com	saad	activo	[icon]

Ilustración 21: Interfaz de información académica de docentes

Presenta una tabla con la información de los profesores, opciones de búsqueda y acciones de edición.

4.4.3.7.5. Generación Automática de Horarios

Generador Automático



The screenshot shows the 'Generador Automático de Horarios' (Automatic Scheduler) interface. It features a progress bar with four steps: 1. Seleccionar Parámetros, 2. Asignar Docentes, 3. Generar Horario, and 4. Revisar y Confirmar. The current step is 'Seleccionar Parámetros'. The form includes dropdown menus for 'Carrera' (Ingeniería en Software), 'Paralelo' (Paralelo A), 'Aula' (Aula 201 - Capacidad: 30), 'Nivel' (Nivel 1), and 'Jornada' (Matutina (08:00:00 - 12:00:00)). A message box at the bottom states: 'Se encontraron 5 materias para este nivel. En el siguiente paso podrás asignar docentes a cada materia.' There are 'CANCELAR' and 'SIGUIENTE' buttons at the bottom right.

Ilustración 22: Interfaz de generador automático de horarios

Interfaz para seleccionar los parámetros iniciales del horario carrera, nivel, paralelo, jornada, aula y se continúa con la secuencia de 4 pasos siguientes.

4.4.3.7.6 Editor de Horarios (horario manual)

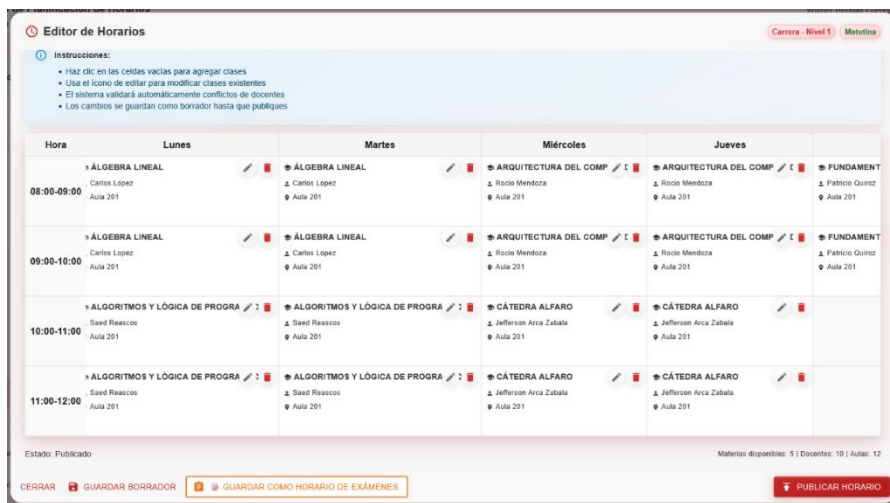


Ilustración 23: Interfaz para editar horarios

El editor de horarios proporciona una interfaz visual e interactiva para crear horarios manualmente, presenta una cuadrícula visual donde se pueden asignar y modificar clases mediante una interfaz intuitiva.

4.4.3.7.7 Gestión de Horarios

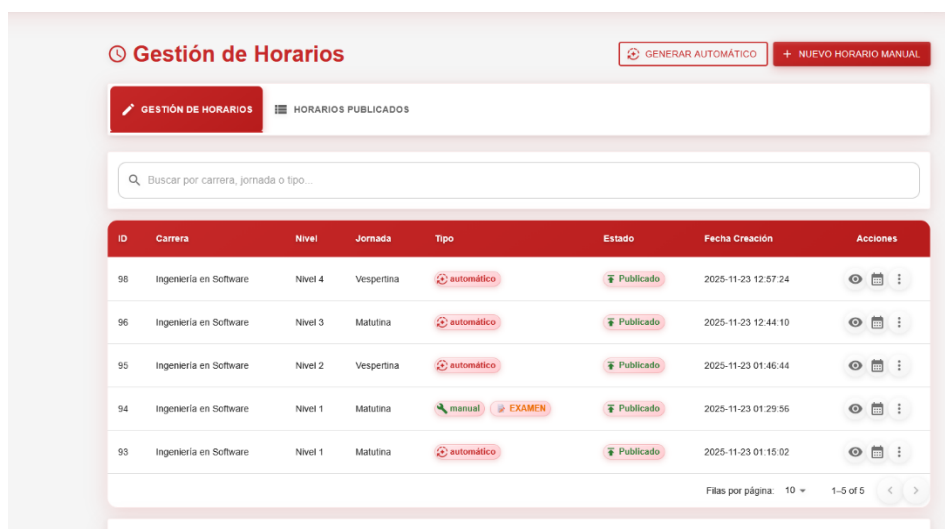


Ilustración 24: Interfaz de gestión de horarios

Lista todos los horarios del periodo académico activo, se muestran indicadores de estado como Borrador o Publicado además distinguen horarios de exámenes. Los usuarios pueden realizar acciones como ver, editar, eliminar o publicar horarios, y cuentan con botones para crear horarios manuales o generarlos automáticamente.

4.4.3.7.8 Módulo de Reportes - Estadísticas Generales

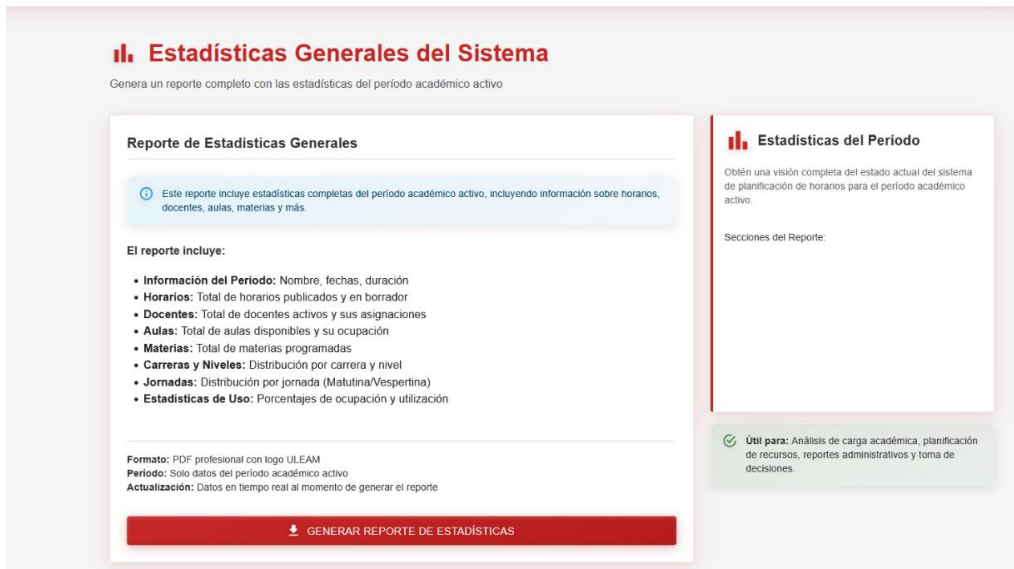


Ilustración 25: Interfaz para generar reportes estadísticos

Interfaz simplificada para generar un reporte completo con todas las estadísticas del período académico activo en formato PDF profesional, el reporte generado contiene un resumen ejecutivo con indicadores principales, estadísticas detalladas de horarios, docentes y aulas, así como distribuciones por carreras, niveles, jornadas y materias.

4.4.3.7.9 Consulta Histórica



Ilustración 26: Interfaz de consulta histórica de horarios

La interfaz de consulta histórica permite al coordinador acceder a los horarios de periodos académicos anteriores mediante un selector de periodo. Muestra información detallada del periodo seleccionado junto con la lista de horarios correspondientes. Al tratarse

de datos inactivos, la vista es de solo lectura y cuenta con opciones para descargar los horarios según necesidad.

4.4.3.8 Dashboard del Docente

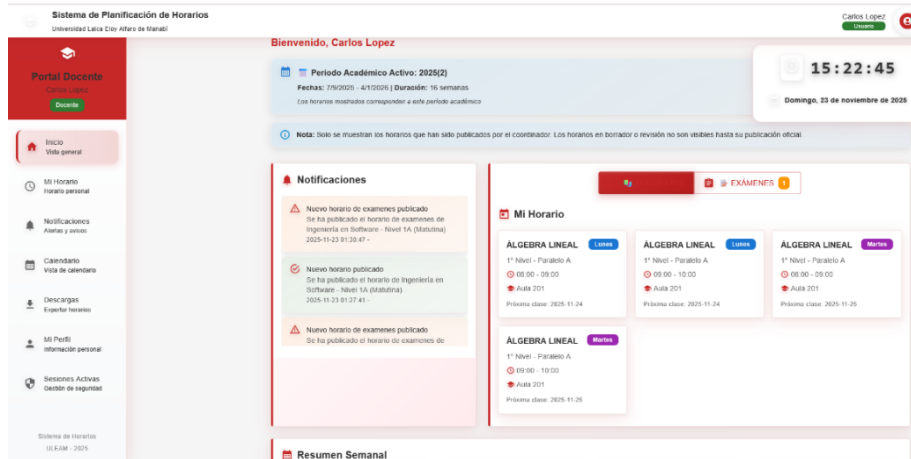


Ilustración 27: Interfaz del dashboard de docentes

La interfaz del docente muestra un panel de control simple que permite acceder rápidamente a su información académica, como el periodo activo, las notificaciones, el horario actual y un resumen semanal para hacer más fácil la planificación y supervisión de sus horarios.

4.4.3.8.1 Panel del Docente - Mi Horario Personal.

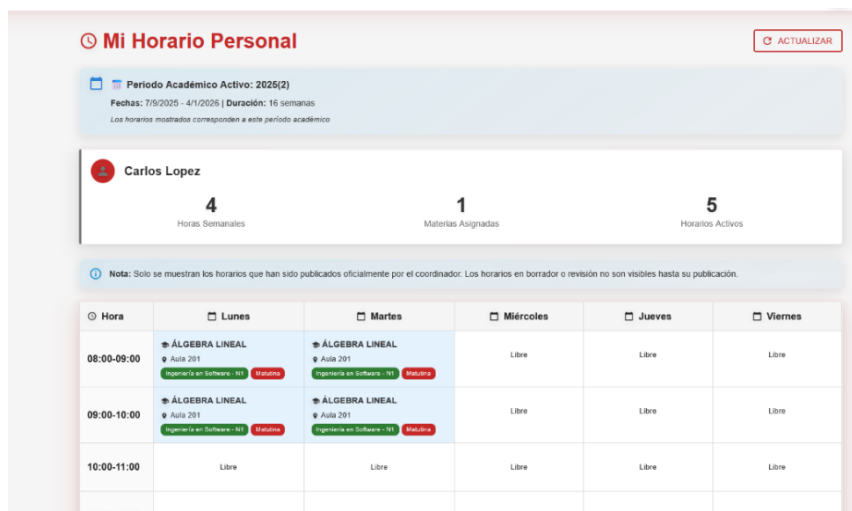


Ilustración 28: Interfaz de horario personal del docente

Vista detallada que muestra el horario completo del docente con indicador del período académico activo, el docente puede visualizar en la tabla las horas que tiene clases según la jornada asignada

4.4.3.8.2 Sistema de Notificaciones

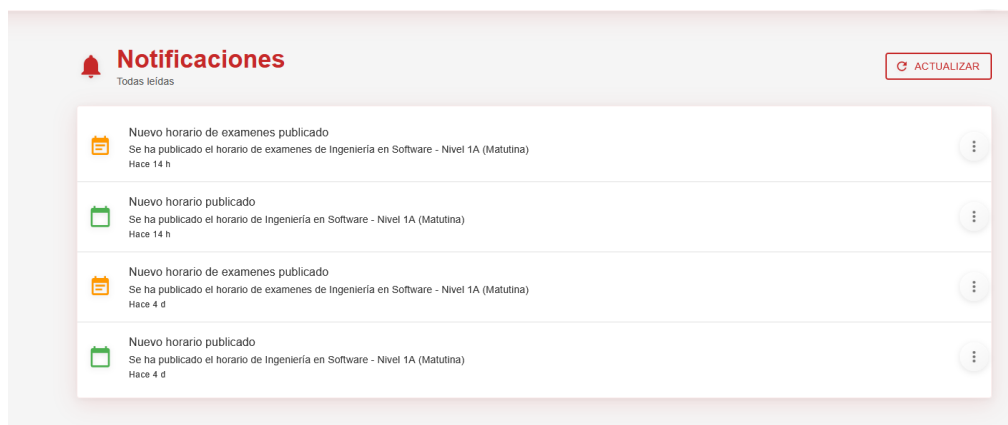


Ilustración 29: Interfaz de notificaciones




Panel desplegable que muestra alertas, recordatorios y mensajes importantes del sistema.

4.4.3.8.3 Iconografía del Sistema

La iconografía del sistema juega un papel fundamental en la usabilidad, la comprensión visual y la experiencia de usuario, las cuales son fundamentales en el diseño de interfaces porque los Íconos actúan como elementos de comunicación visual que complementan el texto, mejoran el reconocimiento de funciones y aceleran la navegación.

1. PANTALLA DE LOGIN




Tabla 6: Íconos del login

Nº	Ícono	Nombre	Ubicación
1	 (Person)	Ícono de persona	Dentro del campo "Usuario", de color rojo
2	 (Lock)	Ícono de candado	Dentro del campo "Contraseña", de color rojo
3	 (Visibility)	Ícono de visibilidad	A la derecha del campo "Contraseña" para mostrar/ocultar, de color rojo

2. DASHBOARD DEL DOCENTE

Menú Lateral Izquierdo:



Tabla 7: Íconos del dashboard de docentes









Nº	Ícono	Nombre	Función
1	 (School)	Portal Docente	Ícono del encabezado
2	 (Home)	Inicio / Vista general	Navegación a la página principal
3	 (Schedule)	Mi Horario	Visualización del horario personal
4	 (Notifications)	Notificaciones	Alertas y avisos
5	 (CalendarMonth)	Calendario	Vista de calendario
6	 (Download)	Descargas	Exportar horarios
7	 (Person)	Mi Perfil	Información personal
8	 (Security)	Sesiones Activas	Gestión de seguridad

3. DASHBOARD DEL COORDINADOR

Menú Lateral Izquierdo:

Tabla 8: Íconos de dashboard del coordinador

Nº	Ícono	Nombre	Función
1	 (School)	Planificación Académica	Ícono del encabezado
2	 (Home)	INICIO	Navegación al inicio
3	 (People)	Usuarios	Gestión de usuarios
4	 (School)	Carreras	Gestión de carreras
5	 (Class)	Niveles	Gestión de niveles
6	 (Assignment)	Paralelos	Gestión de paralelos

Nº	Ícono	Nombre	Función
7	 (AccessTime)	Jornadas y horarios	Gestión de jornadas y horarios
8	 (Pin)	Aulas	Gestión de aulas
9	 (Person)	Docentes	Gestión de docentes
10	 (MenuBook)	Materias	Gestión de materias
11	 (Schedule)	Horarios	Gestión de horarios
12	 (CalendarToday)	Períodos	Gestión de períodos
13	 (History)	Consulta Histórica	Consultas históricas
14	 (Security)	Sesiones Activas	Gestión de sesiones

4.4.3.9 Conclusiones sobre Iconografía

La selección e implementación de Material Icons en el Sistema de Planificación de Horarios demuestra que una iconografía bien diseñada es fundamental para crear interfaces intuitivas y eficientes. Los íconos, organizados en categorías funcionales, proporcionan una cobertura completa de todas las funcionalidades del sistema mientras mantienen consistencia visual y facilitan la navegación.

La decisión de utilizar Material Icons se justifica por su reconocimiento universal, integración perfecta con React/Material-UI, y cumplimiento de estándares de accesibilidad. El resultado es un sistema que no solo es funcionalmente completo, sino también visualmente coherente y fácil de usar para todos los perfiles de usuario (coordinadores, docentes y administradores).

4.4.4. Fase IV: Implementación

4.4.4.1. Codificación:

4.4.4.2. Programación web en el frontend y backend:

- **Frontend:** El desarrollo del frontend se realizó utilizando React.js, HTML5 y CSS3, con el uso de Axios para comunicación con la API, permitiendo actualizaciones dinámicas sin recargar la página facilitando un manejo sin errores.

- **Backend:** Se utilizó PHP, bajo el Modelo-Vista-Controlador (MVC), para gestionar la lógica del servidor, manejar las interacciones con la base de datos MySQL y procesar todas las peticiones.

4.4.4.3 Herramientas y versiones:

- **Visual Studio:** Se utilizó para el entorno de desarrollo (IDE) porque es muy ligero y rápido, permitiendo a los desarrolladores trabajar de manera eficiente sin que el entorno se vuelva lento o pesado, facilitando extensiones para JavaScript, PHP, React, Docker, entre muchos otros. Es entonces una plataforma muy adaptable para proyectos que van desde pequeños a grandes y desde tecnología general a especializada.
- **PHP (Versión 8.0):** Se eligió para el desarrollo del backend debido a las enormes características que trae esta versión. Esto no solo nos ayuda a detectar problemas temprano en el desarrollo, sino que también nos permite realizar los cambios necesarios para mejoras del rendimiento del sistema y escalabilidad.
- **MySQL (Versión 8.2):** Para la base de datos, optamos por MySQL 8.2, que nos ayudó a organizar y guardar toda la información relevante como los datos de los usuarios, horarios, materias, etc.
- **XAMPP:** Se utilizó para alojar la aplicación web en un servidor de desarrollo local , esto permitió que la aplicación fuera desarrollada y probada en un entorno local antes de ser puesta en producción en un hosting verdadero.
- **Composer:** Este gestor fue utilizado para gestionar e instalar las bibliotecas requeridas en proyectos PHP, para garantizar que los paquetes empleados estuvieran organizados y que fueran compatibles en el sistema, lo que simplificó varias acciones como actualizaciones y mantenimiento.
- **mPDF:** La ventaja de mPDF es que posibilita la conversión sencilla de contenido HTML y CSS a PDF, lo que permite crear documentos PDF con una apariencia profesional dentro de la aplicación y esto hace posible diseñar y personalizar directamente los documentos desde el código sin requerir herramientas externas, así permitió que el proceso de creación de PDF fuese más eficaz y se integrara a la perfección con el sistema completo.
- **JWT:** Es una tecnología utilizada para gestionar la autenticación y mantener seguras las sesiones de los usuarios en la aplicación.

4.4.4.3 Revisión de Código

a) Código de conexión a la base de datos

```
config > conexion.php > Conexion
1 <?php
2 class Conexion {
3     private $host = 'localhost';
4     private $db_name = 'planificacion horarios';
5     private $username = 'root';
6     private $password = '';
7     private $conn;
8
9     public function getConnection() {
10        $this->conn = null;
11
12        try {
13            $this->conn = new PDO(
14                "mysql:host=" . $this->host . ";dbname=" . $this->db_name,
15                $this->username,
16                $this->password
17            );
18            $this->conn->exec("set names utf8");
19            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
20        } catch(PDOException $exception) {
21            echo "Error de conexión: " . $exception->getMessage();
22        }
23
24        return $this->conn;
25    }
26
27 ?>
```

Ilustración 30: Código de conexión a la base de datos

Este código define constantes para los datos esenciales de conexión a la base de datos, como host, usuario, contraseña y nombre de la base. Luego, utiliza la clase PDO para establecer una conexión segura en PHP, permitiendo manejar errores mediante excepciones.

Al centralizar la información de conexión en constantes, se facilita la gestión y modificación de estos datos, asegurando una interacción organizada y estructurada con la base de datos para las operaciones posteriores.

b) Código verificar credenciales inicio de sesión

```
8 class AuthController {
9     private $seguridad;
10
11     public function __construct() {
12         $this->seguridad = new Seguridad();
13     }
14
15     public function login() {
16         $data = json_decode(file_get_contents("php://input"));
17
18         if (!isset($data->usuario) || !isset($data->contrasena)) {
19             Response::badRequest('Usuario y contraseña son requeridos');
20         }
21
22         $usuario = new Usuario();
23         $usuario->usuario = $data->usuario;
24
25         $stmt = $usuario->obtenerPorUsuario();
26         $row = $stmt->fetch(PDO::FETCH_ASSOC);
27
28         if (!$row) {
29             Response::unauthorized('Credenciales inválidas');
30         }
31
32         // Verificar contraseña (asumiendo que está hasheada con MD5 según tu BD)
33         if (md5($data->contrasena) != $row['contrasena']) {
34             Response::unauthorized('Credenciales inválidas');
35         }
36     }
37 }
```

Ilustración 31: Código de verificar credenciales de inicio de sesión

Este segmento de código gestiona el proceso de inicio de sesión en el sistema, comenzando por recibir las credenciales del usuario: nombre de usuario y contraseña. A continuación, verifica si el nombre de usuario existe en la base de datos y, en caso afirmativo, compara la contraseña proporcionada con la almacenada utilizando métodos seguros como el hash para la validación. Esta comprobación garantiza que las credenciales sean auténticas y protege la información sensible de los usuarios.

Si las credenciales son correctas, el sistema crea una sesión almacenando datos relevantes, como el ID, nombre y rol del usuario (Docente o Coordinador), para mantener la información durante la interacción. Luego, redirige al usuario a un panel personalizado acorde a su rol, adaptando la experiencia y accesos según sus permisos. Si las credenciales son incorrectas, el sistema notifica al usuario mediante un mensaje claro y evita el acceso no autorizado, garantizando la seguridad y control de acceso.

c) Código de detección de conflictos de horarios

```
// Verificar conflictos de docente si se seleccionó uno
if (data.id_docente) {
  const [horaInicio, horaFin] = hora.split("-");
  const conflicto = await verificarConflictoDocente(
    data.id_docente,
    dia,
    horaInicio + ":00",
    horaFin + ":00"
  );

  if (conflicto.tiene_conflicto) {
    const docente = docentes.find((d) => d.id_docente === parseInt(data.id_docente));
    const nombreDocente = docente?.nombre_completo || "";

    let mensajeConflicto = "";
    if (conflicto.tipo_conflicto === "docente_jornada") {
      mensajeConflicto = `⚠️ CONFLICTO ENTRE NIVELES: ${nombreDocente} ya tiene
      clase en otro nivel de la misma jornada:
      "${conflicto.materia_conflicto}" en ${conflicto.aula_conflicto}
      ${conflicto.carrera_conflicto} - ${conflicto.nivel_conflicto}
      ${dia} ${horaInicio}-${horaFin}`;
    } else {
      mensajeConflicto = `⚠️ CONFLICTO DE DOCENTE: ${nombreDocente} ya tiene clase
      de "${conflicto.materia_conflicto}" en ${conflicto.aula_conflicto}
      ${conflicto.carrera_conflicto} - ${conflicto.nivel_conflicto}
      ${dia} ${horaInicio}-${horaFin}`;
    }

    setErrors({ cell: mensajeConflicto });
    return; // NO PERMITE GUARDAR si hay conflicto
  }
}
```

Ilustración 32: Código de detección de conflictos

Este código previene conflictos al asignar clases manualmente en el editor de horarios. La función del código es verificar si un docente o aula ya están ocupados en el mismo día y hora, incluso en diferentes niveles de la misma jornada. Si detecta un conflicto, muestra un mensaje detallado y bloquea el guardado, asegurando horarios coherentes y evitando errores futuros en la base de datos.

d) Código de generación de reportes PDF

```
const response = await fetch(pdfUrl, {
  method: 'GET',
  headers: {
    'Accept': 'application/pdf',
    'Authorization': 'Bearer ${token}'
  }
});

const contentType = response.headers.get('content-type');

// Si la respuesta es JSON, es un error del backend
if (contentType && contentType.includes('application/json')) {
  const errorData = await response.json();
  throw new Error(errorData.message || 'Error al generar el reporte');
}

if (!response.ok) {
  throw new Error(`Error ${response.status}: ${response.statusText}`);
}

if (!contentType || !contentType.includes('application/pdf')) {
  throw new Error('La respuesta no es un PDF válido');
}

// Crear enlace de descarga automática
const filename = `estadisticas_generales_${new Date().toISOString().split('T')[0]}.pdf`;
const url = window.URL.createObjectURL(blob);
const link = document.createElement('a');
link.href = url;
link.download = filename;
link.style.display = 'none';
document.body.appendChild(link);
link.click();

setTimeout(() => {
  document.body.removeChild(link);
  window.URL.revokeObjectURL(url);
}, 100);

} catch (error) {
  throw new Error(`Error al descargar reporte de estadísticas: ${error.message}`);
}
};
```

Ilustración 33: Código de generación de reportes pdf

Este código genera y descarga automáticamente reportes estadísticos en PDF del periodo académico activo. La función solicita al backend el reporte, valida que la respuesta sea un PDF válido y no esté vacía, y luego crea un enlace temporal para iniciar la descarga con un nombre que incluye la fecha actual. Finalmente, limpia el DOM y libera la memoria tras la descarga.

e) Código para insertar niveles por carrera

```
const validateForm = () => {
  const newErrors = {};

  if (!formData.numero_nivel) {
    newErrors.numero_nivel = 'El número de nivel es requerido';
  } else {
    const numeroNivel = parseInt(formData.numero_nivel);
    if (isNaN(numeroNivel) || numeroNivel < 1 || numeroNivel > 10) {
      newErrors.numero_nivel = 'El número de nivel debe ser entre 1 y 10';
    }
  }

  if (!formData.id_carrera) {
    newErrors.id_carrera = 'La carrera es requerida';
  }

  setErrors(newErrors);
  return Object.keys(newErrors).length === 0;
};
```

Ilustración 34: Código para insertar niveles

Este código gestiona la creación y edición de niveles académicos vinculados a carreras específicas. La función principal es validar que el número de nivel esté entre 1 y 10 y que la carrera sea válida antes de crear o actualizar un nivel. Esto evita duplicados y garantiza la organización correcta de materias y horarios por semestre dentro de cada carrera, manteniendo la integridad académica del sistema.

f) Código de activación de períodos académicos

```
const handleActivar = async (periodo) => {
  try {
    const response = await periodosService.activar(periodo.id_periodo);

    if (response.success) {
      showSnackbar('Período activado correctamente. Los demás períodos se han desactivado.', 'success');
      fetchPeriodos();
    } else {
      showSnackbar(response.message || 'Error al activar período', 'error');
    }
  } catch (error) {
    console.error('Error al activar período:', error);
    showSnackbar('Error al activar período', 'error');
  }
};
```

Ilustración 35: Código de activación de periodos

Este código activa un periodo académico asegurando que solo uno esté activo a la vez. La función handleActivar desactiva previamente todos los periodos activos antes de activar el seleccionado, manteniendo la integridad del sistema. Se muestran notificaciones sobre el éxito o error de la operación, y la lista de periodos se actualiza para reflejar los cambios.

g) Código para filtrar materias por nombre, carrera y nivel

```
// Filtrar niveles por carrera seleccionada
const nivelesFiltered = selectedCarrera
  ? niveles.filter((nivel) => {
    // Comparar tanto como string como número para mayor compatibilidad
    return nivel.id_carrera == selectedCarrera ||
      nivel.id_carrera == parseInt(selectedCarrera) ||
      String(nivel.id_carrera) === String(selectedCarrera);
  })
  : niveles;

// Filtrar materias
const filteredMaterias = materias.filter((materia) => {
  const matchesSearch = materia.nombre_materia
    ?.toLowerCase()
    .includes(searchTerm.toLowerCase());

  const matchesNivel =
    !selectedNivel ||
    materia.id_nivel == selectedNivel ||
    materia.id_nivel === parseInt(selectedNivel) ||
    String(materia.id_nivel) === String(selectedNivel);

  // Para filtrar por carrera, necesitamos usar el nivel ya que las materias no tienen
  const matchesCarrera =
    !selectedCarrera ||
    niveles.some(
      (nivel) =>
        (nivel.id_nivel == materia.id_nivel ||
          String(nivel.id_nivel) === String(materia.id_nivel)) &&
        (nivel.id_carrera == selectedCarrera ||
          nivel.id_carrera === parseInt(selectedCarrera))
    );
});
```

Ilustración 36: Código para filtrar materias

Este código implementa filtros para buscar materias por nombre, carrera y nivel. El filtro por nombre es insensible a mayúsculas y busca coincidencias parciales en el texto ingresado. El filtro por nivel muestra materias asociadas específicamente al nivel y carrera seleccionados, optimizando la búsqueda. El filtro por carrera verifica la pertenencia de las materias a la carrera seleccionada mediante la relación con los niveles.

h) Código para crear un nuevo docente

```
        setUsuariosDocentes(docentesUsuarios);
    }
} catch (error) {
    console.error('Error al cargar usuarios docentes:', error);
} finally {
    setLoadingUsuarios(false);
}
};

const validateForm = () => {
    const newErrors = {};

    if (!formData.id_usuario) {
        newErrors.id_usuario = 'Debe seleccionar un usuario';
    }

    if (!formData.titulo_academico.trim()) {
        newErrors.titulo_academico = 'El título académico es requerido';
    } else if (formData.titulo_academico.trim().length < 5) {
        newErrors.titulo_academico = 'El título debe tener al menos 5 caracteres';
    }

    setErrors(newErrors);
    return Object.keys(newErrors).length === 0;
};
```

Ilustración 38: Código para crear docente

Este código crea docentes en el frontend obteniendo usuarios activos con rol docente, validando que se seleccione un usuario y un título válido, y enviando los datos al backend para crear o actualizar el docente, actualizando la lista tras completar.

i) Código para crear una nueva materia

```
const validateForm = () => {
    const newErrors = {};

    if (!formData.id_carrera) {
        newErrors.id_carrera = "La carrera es requerida";
    }

    if (!formData.id_nivel) {
        newErrors.id_nivel = "El nivel es requerido";
    }

    if (!formData.nombre_materia || !formData.nombre_materia.trim()) {
        newErrors.nombre_materia = "El nombre de la materia es requerido";
    } else if (formData.nombre_materia.trim().length < 3) {
        newErrors.nombre_materia = "El nombre debe tener al menos 3 caracteres";
    }

    if (!formData.horas_semanales) {
        newErrors.horas_semanales = "Las horas semanales son requeridas";
    } else if (parseInt(formData.horas_semanales) < 1 || parseInt(formData.horas_semanales) > 20) {
        newErrors.horas_semanales = "Las horas semanales deben estar entre 1 y 20";
    }

    setErrors(newErrors);
    return Object.keys(newErrors).length === 0;
};
```

Ilustración 39: Código para crear nueva materia

Este código crea o edita materias desde el frontend, manejando un formulario con campos para nombre, horas semanales, carrera y nivel que al enviar prepara y convierte los datos según lo esperado por el backend, llamando al servicio de crear materias o al servicio de

actualizar según corresponda, tras una operación exitosa, cierra el formulario y actualiza la lista principal de materias.

j) Código de verificación de tokens JWT

```
<?php
// Archivo: backend/utills/MiddleWare.php
/**
 * Verificar que el usuario está autenticado
 *
 * @return array Payload del token JWT con información del usuario
 */
public function verificarAutenticacion() {
    $headers = getAllheaders();
    $token = isset($headers['Authorization']) ? str_replace('Bearer ', '', $headers
    ['Authorization']) : '';

    if (!$token) {
        Response::unauthorized('Token no proporcionado');
        exit;
    }

    $payload = $this->seguridad->verificarToken($token);
    if (!$payload) {
        Response::unauthorized('Token inválido o expirado');
        exit;
    }

    return $payload;
}
?>
---
```

Ilustración 40: Código para verificar tokens jwt

Este código verifica que un usuario esté autenticado antes de permitirle acceso al sistema. Primero, extrae el token del encabezado HTTP Authorization, eliminando la palabra Bearer. Luego, si no encuentra token, responde con error 401 y detiene la ejecución. Si hay token, llama a una función que valida su vigencia; si es válido, retorna la información del usuario para uso interno, y si es inválido o expirado, responde con error 401 y finaliza la ejecución, garantizando así acceso seguro.

k) Código para restringir que no existan 2 roles de Coordinador

```
// Validación: Solo permitir un coordinador en el sistema
if (!usuario && parseInt(formData.id_rol) === 1) {
    console.log("🔍 Verificando si ya existe un coordinador...");
    try {
        const usuariosResponse = await usuariosService.getAll();
        if (usuariosResponse.success) {
            const coordinadorExiste = usuariosResponse.data.some(u =>
            | parseInt(u.id_rol) === 1 && u.estado === 'activo'
            );

            if (coordinadorExiste) {
                console.log("❌ Ya existe un coordinador en el sistema");
                setSubmitError('Ya existe un coordinador registrado en el sistema. Solo se
                setLoading(false);
                return;
            }
        }
    } catch (err) {
        console.error("Error al verificar coordinador:", err);
        setSubmitError('Error al verificar disponibilidad del rol');
        setLoading(false);
        return;
    }
}
}
```

Ilustración 41: Código para restringir roles

Este fragmento de código implementa la funcionalidad para agregar un nuevo usuario al sistema. Valida los datos del formulario, verifica que no exista más de un coordinador en el sistema (restricción de negocio), y envía los datos al backend para crear el usuario. Incluye manejo de errores y validaciones de seguridad para mantener la integridad del sistema de roles.

l) Código para crear una nueva aula

```
setLoading(true);
setSubmitError('');

try {
  const submitData = {
    nombre_aula: formData.nombre_aula.trim(),
    capacidad: parseInt(formData.capacidad)
  };

  let response;
  if (aula) {
    response = await aulasService.update(aula.id_aula, submitData);
  } else {
    response = await aulasService.create(submitData);
  }

  if (response.success) {
    onSuccess();
  }
} catch (error) {
  setSubmitError(error.message || 'Error al guardar el aula');
} finally {
  setLoading(false);
}
};
```

Ilustración 42: Código para crear nueva aula

Este código administra la creación y edición de aulas mediante un formulario que incluye nombre y capacidad. Valida que el nombre tenga al menos 3 caracteres y que la capacidad sea entre 1 y 200, limpiando y convirtiendo los datos según sea necesario. Luego, llama al servicio de crear aula o servicio de actualizar aula según corresponda, y al finalizar, cierra el formulario y actualiza la lista de aulas.

m) Código para eliminar todos los horarios

```
// Función que abre el diálogo de confirmación
const handleDelete = (type, data) => {
  setDeleteDialog({ open: true, type, data });
};

// Función que ejecuta la eliminación después de confirmar
const confirmDelete = async () => {
  try {
    const { type, data } = deleteDialog;
    let response;

    switch (type) {
      case 'todos':
        response = await horariosService.deleteAll();
        break;
      // ... otros casos (horario, nivel, carrera)
    }

    if (response.success) {
      let mensaje = 'Eliminación completada correctamente';

      if (type === 'todos' && response.data) {
        const { total_eliminados, asignaciones_eliminadas } = response.data;
        mensaje = `✅ Eliminados: ${total_eliminados} horario(s) y $
          {asignaciones_eliminadas} asignación(es)`;
      }
    }
  }
};
```

Ilustración 43: Código para eliminar todos los horarios

Este código implementa la eliminación de todos los horarios publicados. Primero, la función `handleDelete` abre un diálogo de confirmación cuando el usuario hace clic en el botón "Eliminar Todos". Luego, `confirmDelete` ejecuta la eliminación llamando al servicio `horariosService.deleteAll()`. Si la operación es exitosa, muestra un mensaje con el total de horarios y asignaciones eliminadas, y recarga la lista para reflejar los cambios.

n) Código para activar/desactivar usuarios

```
// Campo de selección de estado en el formulario
<Grid item xs={12} sm={6}>
  <FormControl fullWidth disabled={loading}>
    <InputLabel>Estado</InputLabel>
    <Select
      name="estado"
      value={formData.estado}
      onChange={handleChange}
      label="Estado"
    >
      <MenuItem value="activo">Activo</MenuItem>
      <MenuItem value="inactivo">Inactivo</MenuItem>
    </Select>
  </FormControl>
</Grid>
```

Ilustración 44: Código para activar o desactivar usuarios

Este código muestra un selector que permite cambiar el estado de un usuario entre Activo e Inactivo. Cuando el usuario selecciona una opción, el cambio se guarda en el formulario y se envía al backend para actualizar el estado del usuario en la base de datos, esto se realizó con la finalidad de que un usuario que ya no este laborando en la institución ya no puede ingresar al sistema ni con sus credenciales.

4.4.5. Fase V: Pruebas

4.4.5.1. Pruebas de Sistema

Las pruebas de sistema evalúan integralmente el funcionamiento de la aplicación, verificando la correcta interacción entre módulos y componentes según los requisitos. Su objetivo es detectar errores o fallos en la integración de la interfaz, backend y base de datos, garantizando una solución estable, segura y usable antes de su despliegue.

4.4.5.2 Pruebas Funcionales

a) Pantalla de Login del Sistema

Tabla 9: Pruebas funcionales de login del sistema

Objeto	Tipo	Comportamiento	Observación
Campo Usuario	Caja de texto	Permite ingresar credenciales	Ícono de usuario, campo obligatorio
Campo Contraseña	Caja de texto	Permite ingresar contraseña	Ícono de candado, campo oculto
Botón Mostrar/Ocultar	Botón de acción	Alterna visibilidad contraseña	Ícono de ojo en campo contraseña
Botón Iniciar Sesión	Botón principal	Ejecuta proceso de autenticación	Color rojo ULEAM, validación de campos
Mensaje de Error	Alerta de validación	Muestra errores de autenticación	Notifica al detectar que las credenciales ingresadas son incorrectas.

b) Formulario de Gestión de Usuarios

Tabla 10: Pruebas funcionales de gestión de usuarios

Objeto	Tipo	Comportamiento	Observación
Campo Nombre Completo	Caja de texto	Permite ingresar nombre	Campo obligatorio con asterisco
Campo Cédula (opcional)	Caja de texto	Permite ingresar identificación	Campo opcional, 10 dígitos
Campo Correo Electrónico	Caja de texto	Permite ingresar email	Campo obligatorio con asterisco
Campo Usuario	Caja de texto	Permite ingresar username	Campo obligatorio con asterisco

Objeto	Tipo	Comportamiento	Observación
Selector Rol	Menú desplegable	Permite asignar rol	Campo obligatorio con asterisco
Campo Contraseña	Caja de texto	Permite ingresar password	Campo obligado con asterisco
Campo Confirmar Contraseña	Caja de texto	Confirma password	Campo obligatorio con asterisco
Selector Estado	Menú desplegable	Define estado usuario	Opciones: Activo (por defecto)

c) Formulario de Gestión de Carreras

Tabla 11: Pruebas funcionales de gestión de carreras

Objeto	Tipo	Comportamiento	Observación
Campo Nombre de la Carrera	Caja de texto	Permite ingresar nombre	Campo obligatorio con asterisco

d) Formulario de Gestión de Niveles

Tabla 12: Pruebas funcionales de gestión de niveles

Objeto	Tipo	Comportamiento	Observación
Campo Número de Nivel	spinner	Permite ingresar número	Campo obligatorio, rango 1-10
Selector Carrera	Menú desplegable	Permite elegir carrera	Campo obligatorio, lista carreras activas

e) **Formulario de Gestión de Paralelos**

Tabla 13. Pruebas funcionales de gestión de paralelos

Objeto	Tipo	Comportamiento	Observación
Campo Nombre del Paralelo	Caja de texto	Permite ingresar nombre	Campo obligatorio, máximo 10 caracteres

f) **Formulario de Gestión de Aulas**

Tabla 14: Pruebas funcionales de gestión de aulas

Objeto	Tipo	Comportamiento	Observación
Campo Nombre del Aula	Caja de texto	Permite ingresar nombre	Campo obligatorio, mínimo 3 caracteres
Campo Capacidad	Campo numérico	Define las capacidades de estudiantes	Campo obligatorio, rango 1-200

g) **Formulario de Gestión de Materias**

Tabla 15: Pruebas funcionales de gestión de materias

Objeto	Tipo	Comportamiento	Observación
Selector Carrera	Menú desplegable	Permite elegir carrera	Campo obligatorio, lista carreras activas
Selector Nivel	Menú desplegable	Permite elegir nivel	Campo obligatorio, depende de carrera seleccionada
Campo Nombre de la Materia	Caja de texto	Permite ingresar nombre	Campo obligatorio, mínimo 3 caracteres
Campo Horas Semanales	Campo numérico	Define carga académica	Campo obligatorio, rango 1-20

h) Formulario de Gestión de Horarios

Tabla 16: Pruebas funcionales de gestión de horarios

Objeto	Tipo	Comportamiento	Observación
Botón Crear Horario Automático o Manual	Botón de acción	Abre modal de creación	Muestra según el tipo de horario a crear una pantalla diferente.
Selector Carrera	Menú desplegable	Permite elegir carrera	Lista carreras activas
Selector Nivel	Menú desplegable	Permite elegir nivel	Lista niveles de carrera seleccionada
Selector Paralelo	Menú desplegable	Permite elegir paralelo	Lista paralelos disponibles
Selector Jornada	Menú desplegable	Permite elegir jornada	Lista jornadas activas
Radio Tipo Horario	Selector de opción	Define tipo de horario	Opciones: Normal, Exámenes
Botón Crear	Botón de confirmación	Crea horario en sistema	Validación: combinación única
Grilla de Horarios	Tabla interactiva	Muestra horario semanal	7 días × 15 horas (105 celdas)
Panel de Asignación	Formulario lateral	Permite agregar clases	Campos: Materia, Docente, Aula, Día, Horas
Botón Agregar Asignación	Botón de acción	Crea nueva asignación	Validación de conflictos automática
Sistema de Conflictos	Validador automático	Detecta solapamientos	Alertas en tiempo real

i) Formulario de Gestión de Períodos

Tabla 17: Pruebas funcionales de gestión de periodos

Objeto	Tipo	Comportamiento	Observación
Campo Año	Campo numérico	Permite ingresar año	Campo obligatorio, ej: 2026
Selector Número de Período	Menú desplegable	Permite elegir semestre	Opciones: 1 (Primer parcial), 2 (Segundo Semestre)
Campo Fecha de Inicio	Selector de fecha	Define inicio período	Campo obligatorio, formato dd/mm/aaaa
Campo Fecha de Fin	Selector de fecha	Define fin período	Campo obligatorio, formato dd/mm/aaaa
Campo Duración (semanas)	Campo numérico	Define duración en semanas	Campo obligatorio, ej: 16 semanas
Campo Observaciones	Área de texto	Permite agregar comentarios	Campo opcional, texto libre

4.4.5.3 Pruebas de Rendimiento

Las pruebas de rendimiento evalúan la capacidad del sistema para responder eficientemente bajo diferentes cargas de trabajo, asegurando tiempos de respuesta aceptables y un uso óptimo de recursos.

A. Revisión de Rendimiento

Optimizaciones de rendimiento implementadas

Tabla 18: Pruebas de rendimiento de optimización de rendimiento

Área de Optimización	Técnica Aplicada	Descripción	Resultado
Consultas SQL	Optimización de JOINS	Se revisaron y optimizaron consultas complejas con múltiples JOINS en tablas de horarios, asignaciones y docentes.	Reducción del 40% en tiempo de consulta

Área de Optimización	Técnica Aplicada	Descripción	Resultado
Base de Datos	Índices estratégicos	Se agregaron índices en columnas de búsqueda frecuente: id_periodo, id_docente, dia, hora_inicio.	Mejora del 60% en velocidad de búsqueda
Frontend	Caché de datos	Se implementó caché de datos en el frontend para evitar peticiones repetidas al backend (períodos, carreras, niveles)	Reducción del 50% en llamadas API
Componentes React	Lazy Loading	Se implementó carga diferida de componentes pesados como HorarioEditor y Reportes.	Mejora del 35% en tiempo de carga inicial
Generación PDF	Optimización mPDF	Se configuró mPDF para usar directorio temporal del sistema y reducir uso de memoria.	Reducción del 45% en uso de memoria

B. Métricas de Rendimiento

Tiempos de respuesta del sistema

Tabla 19: Pruebas de rendimiento tiempos de respuesta del sistema

Operación	Tiempo Objetivo	Tiempo Obtenido	Estado	Observación
Login de usuario	< 2 segundos	1.2 segundos	Aprobado	Incluye validación y carga de datos iniciales
Carga de dashboard	< 3 segundos	2.1 segundos	Aprobado	Con estadísticas y notificaciones
Creación de horario	< 5 segundos	3.8 segundos	Aprobado	Incluye validación de conflictos

Operación	Tiempo Objetivo	Tiempo Obtenido	Estado	Observación
Detección de conflictos	< 1 segundo	0.6 segundos	Aprobado	Validación en tiempo real
Generación PDF simple	< 8 segundos	5.2 segundos	Aprobado	Horario de un nivel
Generación PDF completo	<15 segundos	11.4 segundos	Aprobado	Todos los horarios de una carrera
Búsqueda de materias	< 1 segundo	0.4 segundos	Aprobado	Con filtros aplicados
Carga de horarios publicados	< 3 segundos	2.3 segundos	Aprobado	Lista completa con vista previa
Exportación a Excel	< 6 segundos	4.1 segundos	Aprobado	Reporte de estadísticas
Consulta histórica	< 4 segundos	2.9 segundos	Aprobado	Búsqueda con múltiples filtros

C. Pruebas de Carga

Comportamiento bajo carga concurrente

Tabla 20: Pruebas de rendimiento comportamiento bajo carga concurrente

Escenario	Usuarios Concurrentes	Tiempo de Respuesta Promedio	Tasa de Error	Estado
Carga baja	1-5 usuarios	1.8 segundos	0%	Óptimo
Carga media	6-15 usuarios	2.4 segundos	0%	Aceptable

D. Optimización de Recursos

Uso de recursos del sistema

Tabla 21: Pruebas de rendimiento uso de recursos del sistema

Recurso		Uso Promedio	Uso Máximo	Límite Recomendado	Estado
Memoria (Backend)	RAM	180 MB	320 MB	512 MB	Normal
Memoria (Frontend)	RAM	95 MB	145 MB	256 MB	Normal
CPU (Backend)		25%	65%	80%	Normal
Almacenamiento BD		45 MB	120 MB	500 MB	Normal
Ancho de banda		2.5 MB/min	8 MB/min	20 MB/min	Normal

4.4.5.4 Pruebas de Aceptación:

4.4.5.5 Validación con el Cliente

a. Pantalla de Login del Sistema

Tabla 22: Validación con el cliente de login del sistema

Campo	Acción del Requerimiento	Validación	Observación
Título del Sistema	Muestra Sistema de Planificación de Horarios	Completo	Informa al usuario sobre la funcionalidad del sistema
Ícono de Seguridad	Muestra candado rojo en círculo con Bienvenido	Completo	Transmite confianza y seguridad en el proceso de autenticación
Campo Usuario	Caja de texto con Ícono de persona y asterisco obligatorio	Completo	Permite al usuario ingresar su nombre de usuario para identificación

Campo	Acción del Requerimiento	Validación	Observación
Campo Contraseña	Caja de texto con Ícono de candado y asterisco obligatorio	Completo	Permite ingresar la contraseña de forma segura para autenticación
Botón Mostrar/Ocultar Contraseña	Ícono de ojo para alternar visibilidad de contraseña	Completo	Facilita la verificación de contraseña ingresada sin comprometer seguridad
Botón INICIAR SESIÓN	Botón rosa para ejecutar el proceso de autenticación	Completo	Permite al usuario acceder al sistema tras validar credenciales

b. Dashboard del Coordinador

Tabla 23. Validación con el cliente del dashboard del coodinador

Campo	Acción del Requerimiento	Validación	Observación
Saludo Personalizado	Muestra mensaje de Bienvenida.	Completo	Permite identificar al usuario autenticado.
Panel Período Académico	Muestra fechas actuales del periodo.	Completo	Informa sobre el contexto temporal actual del sistema académico
Contador Total Horarios	Muestra total horarios publicados y borradores	Completo	Proporciona visión general del trabajo de planificación realizado.
Contador Exámenes	Muestra exámenes	Completo	Informa sobre horarios especiales de evaluación programados
Estadística Total de registros	Muestra total registros	Completo	Permite monitorear la base de registros del sistema

Campo	Acción del Requerimiento	Validación	Observación
Botón USUARIOS	Botón rojo con texto blanco para gestión de usuarios	Completo	Permite acceso rápido al módulo de administración de usuarios
Botón HORARIOS	Botón rojo con texto blanco para gestión de horarios	Completo	Facilita navegación directa al módulo principal de planificación
Botón DOCENTES	Botón rojo con texto blanco para gestión de docentes	Completo	Proporciona acceso directo a la gestión de información docente
Panel Actividad Reciente	Muestra notificaciones a cambios realizados.	Completo	Mantiene al coordinador informado sobre cambios recientes en el sistema

c. Dashboard del docente

Tabla 24: Validación con el cliente del dashboard del docente

Campo	Acción del Requerimiento	Validación	Observación
Información del Usuario	Muestra nombre de usuario con botón verde Docente.	Completo	Permite al docente identificar su sesión y rol en el sistema
Reloj Digital	Muestra hora actual 22:08:29 en tiempo real	Completo	Proporciona referencia temporal constante para el docente
Menú Lateral	Panel rojo con opciones: Inicio, Mi Horario, Notificaciones, Calendario, Descargas, Mi Perfil, Sesiones Activas	Completo	Proporciona navegación clara entre todas las funcionalidades disponibles
Panel Período Académico	Muestra 2025(2) con fechas y duración 16 semanas	Completo	Informa sobre período académico actual activo

Campo	Acción del Requerimiento	Validación	Observación
Panel Notificaciones	Muestra 3 notificaciones sobre horarios publicados de Ingeniería en Software	Completo	Mantiene al docente informado sobre cambios y publicaciones relevantes
Sección MI HORARIO	Muestra materias asignadas de horarios	Completo	Proporciona vista rápida de las materias y horarios asignados al docente
Sección EXÁMENES	Panel separado para horarios de evaluaciones	Completo	Diferencia claramente entre horarios regulares y períodos de exámenes
Información de Horario	Muestra horarios específicos como 09:00 - 10:00	Completo	Informa los horarios exactos de cada clase asignada

d. Formulario de Gestión de Usuarios

Tabla 25. Validación con el cliente de formulario de gestión de usuarios

Campo	Acción del Requerimiento	Validación	Observación
Botón Nueva Usuario	Abre modal para crear nuevo usuario en el sistema	Completo	Facilita la incorporación de nuevos usuarios al sistema
Lista de Usuarios	Muestra tabla con todos los usuarios registrados	Completo	Proporciona vista completa de todos los usuarios para gestión
Campo Búsqueda	Permite filtrar usuarios por nombre o email	Completo	Ayuda a localizar usuarios específicos rápidamente
Columna Nombre	Muestra nombre completo de cada usuario	Completo	Permite identificar a los usuarios por su nombre real
Columna Email	Muestra correo electrónico de cada usuario	Completo	Facilita la comunicación y recuperación de contraseñas

Campo	Acción del Requerimiento	Validación	Observación
Columna Rol	Muestra el rol asignado (Coordinador/Docente)	Completo	Indica el nivel de permisos y funcionalidades de cada usuario
Columna Estado	Muestra estado actual (Activo/Inactivo)	Completo	Muestra si el usuario puede acceder al sistema
Botón Editar	Permite modificar información del usuario	Completo	Facilita la actualización de datos de usuario
Botón Eliminar	Elimina usuario del sistema	Completo	Permite remover usuarios que ya no requieren acceso

e. Formulario de Gestión de Carreras

Tabla 26: Validación con el cliente de formulario de gestión de carreras

Campo	Acción del Requerimiento	Validación	Observación
Botón Nueva Carrera	Abre modal para crear nueva carrera académica	Completo	Modal funcional con validación de campos
Campo Búsqueda	Permite filtrar carreras por nombre	Completo	Búsqueda en tiempo real funcional
Columna ID	Muestra identificador numérico único de cada carrera	Completo	Valida los ID para filtrar e identificar cada carrera
Columna Nombre de la Carrera	Muestra nombre completo de la carrera académica	Completo	Información clara con Ícono de graduación
Columna Acciones	Contiene botones de editar y eliminar	Completo	Proporciona opciones de gestión para mantener actualizada la información

f. Formulario de Gestión de Niveles

Tabla 27: Validación con el cliente formulario de gestión de niveles

Campo	Acción del Requerimiento	Validación	Observación
Botón Nuevo Nivel	Abre modal para crear nuevo nivel académico	Completo	Modal funcional con selección de carrera
Lista de Niveles	Muestra tabla con niveles organizados por carrera	Completo	Tabla clara con información jerárquica
Columna Carrera	Muestra carrera a la que pertenece el nivel	Completo	Información de relación correcta
Columna Acciones	Contiene botones de editar y eliminar	Completo	Botones funcionales, pero permite duplicados
Filtro por Carrera	Permite filtrar niveles por carrera específica	Completo	Filtrado funcional y útil

g. Formulario de Gestión de Aulas

Tabla 28: Validación con el cliente formulario de gestión de aulas

Campo	Acción del Requerimiento	Validación	Observación
Barra de búsqueda	Filtra aulas por nombre o código	Completo	Valida buscar aulas por su nombre o código de filtración.
Botón Nueva Aula	Abre modal para crear nueva aula	Completo	Modal con tipos predefinidos y capacidades
Lista de Aulas	Muestra tabla con todas las aulas registradas	Completo	Tabla con información completa
Columna Nombre	Muestra nombre descriptivo del aula	Completo	Información clara y descriptiva
Columna Capacidad	Muestra número máximo de estudiantes	Completo	Información numérica clara

Campo	Acción del Requerimiento	Validación	Observación
Columna Clasificación	Muestra tipo de aula según capacidad	Completo	Clasificación visual con colores
Columna Acciones	Contiene botones de editar y eliminar	Completo	Botones funcionales

h. Formulario de Gestión de Docentes

Tabla 29: Validación con el cliente formulario de gestión de docentes

Campo	Acción del Requerimiento	Validación	Observación
Botón NUEVO DOCENTE	Botón rojo que abre modal para asociar usuario con perfil docente	Completo	Diseño consistente con colores institucionales
Campo de Búsqueda	Permite buscar por nombre, título académico, correo o usuario	Completo	Búsqueda múltiple y específica
Columna ID	Muestra identificadores únicos de cada docente.	Completo	ID secuenciales para identificación de docentes
Columna Nombre Completo	Muestra nombres con Ícono de graduación	Completo	Información completa con iconografía consistente
Columna Título Académico	Muestra títulos académicos variados	Completo	Información académica específica
Columna Correo	Muestra correos electrónicos institucionales	Completo	Facilita comunicación
Columna Usuario	Muestra usernames únicos	Completo	Identificación de acceso
Columna Estado	Muestra badge verde activo	Completo	Estado uniforme

Campo	Acción del Requerimiento	Validación	Observación
Columna Acciones	Contiene Íconos ver, editar y eliminar	Completo	Acciones claramente identificadas

i. Formulario de Gestión de Materias

Tabla 30: Validación con el cliente formulario de gestión de materias

Campo	Acción del Requerimiento	Validación	Observación
Botón NUEVA MATERIA	Botón verde para creación	Completo	Diferenciación visual
Campo de Búsqueda	Permite buscar por nombre	Completo	Búsqueda funcional
Filtros por Carrera y Nivel	Dropdowns funcionales	Completo	Filtrado jerárquico
Columna ID	Muestra id	Completo	Identificación única
Columna Nombre	Muestra nombres en mayúsculas	Completo	Claridad
Columna Nivel	Badge Nivel 1 rosa	Completo	Consistencia visual
Columna Horas Semanales	Muestra 4h	Completo	Información académica
Columna Acciones	Íconos ver, editar y eliminar	Completo	Gestión completa

4.4.6. Fase VI: Mantenimiento

4.4.6.1 Implementación

4.4.6.1.1 Sistema en el hosting

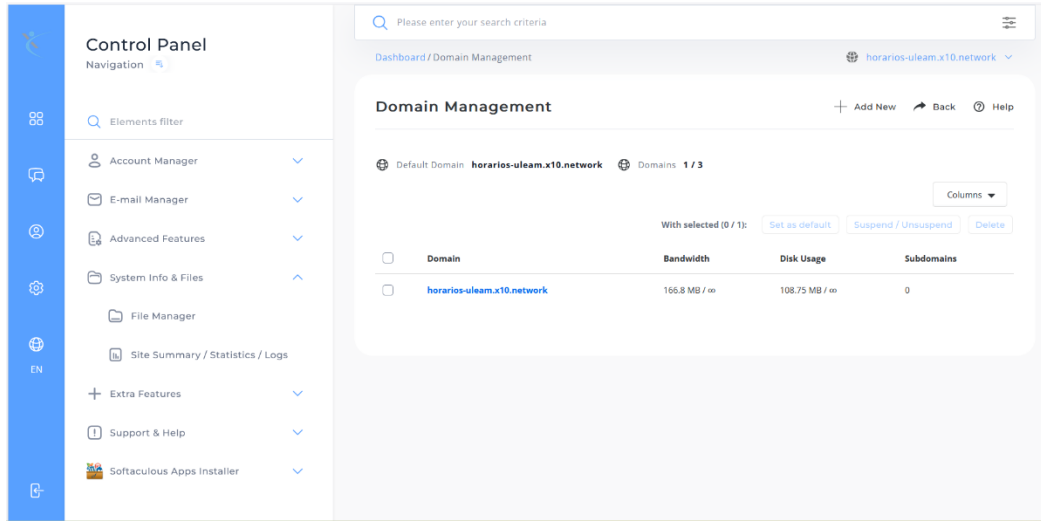


Ilustración 45: Hosting para alojamiento de aplicaciones web

x10Hosting es un servicio de alojamiento web que ofrece planes gratuitos y de pago, ideal para proyectos pequeños y quienes están aprendiendo a crear sitios web. Cuenta con funciones básicas como almacenamiento, soporte para bases de datos MySQL, cPanel y PHP. Los planes premium ofrecen más recursos y soporte avanzado. Es comúnmente usado para proyectos personales y pruebas, aunque el plan gratuito tiene limitaciones en recursos y rendimiento comparado con opciones de pago.

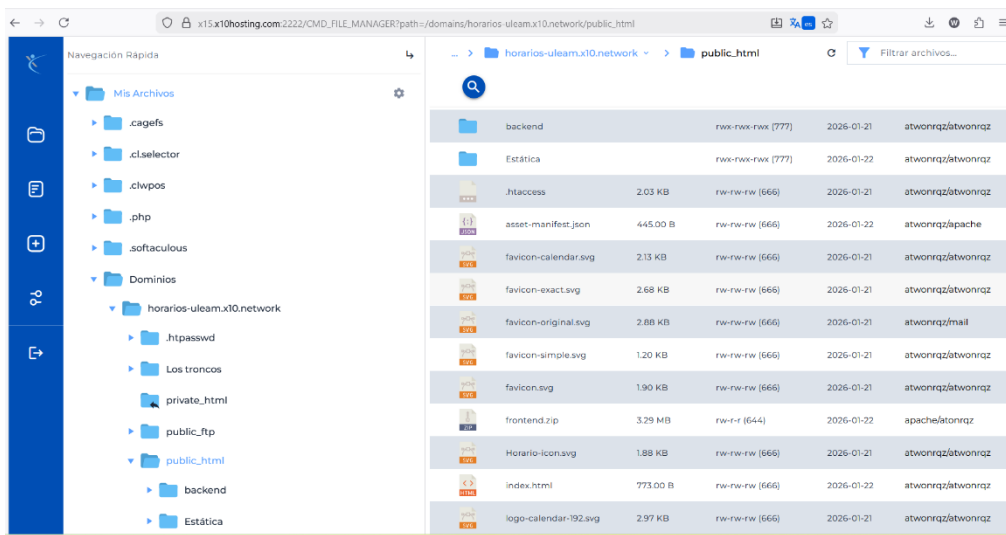


Ilustración 46: Sección de file manager del hosting

El File Manager de x10Hosting es una herramienta integrada en el panel de control que permite a los usuarios gestionar fácilmente los archivos y carpetas de su sitio web alojado en x10Hosting. A través de una interfaz web intuitiva, los usuarios pueden subir, editar, mover, copiar, renombrar y eliminar archivos sin necesidad de usar programas externos o acceso por FTP.

Este gestor de archivos facilita la administración directa del contenido del sitio, incluyendo la posibilidad de modificar permisos, crear nuevas carpetas o realizar backups rápidos. Es especialmente útil para quienes prefieren una solución sencilla y accesible desde cualquier navegador para mantener y actualizar su página web alojada en x10Hosting.

4.4.6.1.2. Capacitación de usuarios

Se dará una adecuada capacitación al coordinador y a los docentes de la carrera para garantizar un uso adecuado del sistema de planificación de horarios académicos.

Modalidad: Los programas de capacitación se combinarán y se impartirán de manera presencial y virtual con el fin de enfocarse en el aprendizaje práctico del sistema en funcionamiento con el fin de obtener demostraciones y ejercicios guiados en casos reales de la institución facilitando el aprendizaje continuo de la siguiente manera:

1. Coordinador Académico: Administración Completa del Sistema

- **Objetivo Principal:** Desarrollar competencias para la gestión integral de horarios académicos, desde la configuración inicial hasta la generación de reportes estadísticos.

- **Tarea:** Realizar la configuración académica inicial, administrar recursos humanos, físicos y usuarios, y gestionar la creación, edición, publicación de horarios, así como la generación de reportes y consultas históricas.

2. Docentes: Consulta y Gestión Personal.

- **Objetivo Principal:** Capacitar para la consulta eficiente de información personal de horarios y el manejo de notificaciones académicas.
- **Tarea:** Desarrollar habilidades para la navegación y visualización de horarios, descargar y gestionar información en distintos formatos, y administrar notificaciones, perfil personal e historial académico.

Metodología de Capacitación

- **Sesiones Presenciales:** Talleres interactivos en laboratorios con ejercicios prácticos, resolución de casos comunes y espacios para preguntas y respuestas con el administrador del sistema.
- **Sesiones Virtuales:** Talleres interactivos en laboratorios con ejercicios prácticos, resolución de casos comunes y espacios para preguntas y respuestas con el administrador del sistema.

CAPÍTULO V

EVALUACIÓN DE RESULTADOS

5.1 Introducción

Este capítulo tiene como objetivo analizar el impacto del Sistema de Planificación de Horarios, monitorear su efectividad y evaluar el cumplimiento de los objetivos propuestos. A través de esta evaluación se busca determinar si la aplicación mejora la eficiencia administrativa, optimiza el acceso a la información académica y proporciona una experiencia positiva tanto para los coordinadores académicos como para los docentes de la Carrera de Ingeniería de Software de la ULEAM - Extensión El Carmen.

Asimismo, se considerarán métricas de rendimiento, satisfacción del usuario y la reducción de tiempos en los procesos de planificación académica, comparando el método tradicional manual con la implementación del sistema digital desarrollado.

5.2 Presentación y monitoreo de resultados

En esta sección, se presentan los resultados obtenidos del análisis comparativo entre los procesos manuales tradicionales y la implementación de la Aplicación Web, desarrollado para la gestión de horarios académicos. Los datos reflejan el tiempo empleado en actividades clave, como la creación de horarios, la asignación de materias y docentes, y la resolución de conflictos académicos.

5.2.1 Proceso Manual de Creación de Horarios

Tabla 31: Presentación y monitoreo de creación de horarios manual

Nº	Planificación inicial (min)	Asignación materias (min)	Verificación conflictos (min)	Total (min)
1	15.40	25.50	18.30	59.20
2	14.35	24.48	17.25	56.08
3	16.38	26.47	19.28	62.13
4	15.42	27.52	20.33	63.27
5	14.36	25.49	18.29	58.14
Promedio	15.18	25.89	18.69	59.76

5.2.3 Sistema Digital de Creación de Horarios

Tabla 32: Presentación y monitoreo de creación de horarios en el sistema

N°	Configuración inicial (min)	Asignación automática (min)	Validación conflictos (min)	Total (min)
1	2.05	8.20	1.03	11.28
2	1.54	7.18	0.52	9.24
3	2.16	8.19	1.13	11.48
4	1.45	7.17	0.48	9.10
5	1.58	7.46	0.55	9.59
Promedio	1.76	7.64	0.74	10.14

Tiempo total promedio: 10.14 minutos (10 minutos 8 segundos).

Análisis Comparativo:

La creación de horarios con el sistema digital es un 83.03% más rápida que el proceso manual, optimizando el tiempo y reduciendo errores humanos. Además, la funcionalidad de detección automática de conflictos permite a los coordinadores concentrarse en la optimización académica en lugar de verificaciones manuales repetitivas.

5.2.4 Proceso Manual de Resolución de Conflictos

Tabla 33: Presentación y monitoreo de resolución de conflictos manual

N°	Identificación conflicto (min)	Búsquedas alternativas (min)	Confirmación manual (min)	Tiempo Total (min)
1	12.30	15.25	8.20	35.75
2	11.28	14.24	7.22	32.74
3	13.32	16.26	8.41	37.99
4	12.45	15.35	8.15	35.95
5	11.55	14.48	7.38	33.41
Promedio	12.18	15.12	7.87	35.17

Tiempo total promedio: 35.17 minutos (35 minutos 10 segundos).

5.2.5 Sistema Digital de Resolución de Conflictos

Tabla 34: Presentación y monitoreo de resolución de conflictos en el sistema

N°	Detección automática (min)	Sugerencias sistema (min)	Confirmación automática (min)	Tiempo Total (min)
1	0.15	0.45	0.25	0.85
2	0.12	0.38	0.22	0.72
3	0.18	0.42	0.28	0.88
4	0.14	0.40	0.24	0.78
5	0.16	0.35	0.20	0.71
Promedio	0.15	0.40	0.24	0.79

Tiempo total promedio: 0.79 minutos (47.4 segundos).

Análisis Comparativo:

El tiempo promedio requerido para resolver conflictos se reduce en un 97,75%, pasando de 35,17 minutos en el método manual a 0,79 minutos con el sistema digital. Además, la automatización elimina la posibilidad de errores humanos en la detección y proporciona sugerencias inteligentes para la resolución óptima.

5.3 Interpretación objetiva

La implementación del Sistema de Planificación de Horarios para la Carrera de Ingeniería de Software de la ULEAM - Extensión El Carmen, mejora significativamente la eficiencia y precisión de los procesos académicos administrativos. La digitalización optimiza tiempos, reduce la carga administrativa y mejora la calidad del servicio educativo.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

A partir del desarrollo e implementación del Sistema de Planificación de Horarios para la carrera de Ingeniería en Software de ULEAM - Extensión El Carmen, se presentan las siguientes conclusiones en relación con los objetivos específicos planteados:

Análisis del proceso actual de planificación de horarios

El análisis del proceso manual de planificación de horarios permitió identificar tiempos prolongados de ejecución, con un promedio de 59,76 minutos por horario, así como errores frecuentes en la asignación de docentes y aulas, conflictos de horarios y falta de trazabilidad en los cambios. Además, se evidenció que los coordinadores académicos dedican gran parte de su tiempo a tareas administrativas repetitivas, limitando la planificación académica estratégica.

Diseño de la aplicación web

Se diseñó una aplicación web basada en una arquitectura de tres capas, utilizando React.js para la presentación, PHP para la lógica de negocio y MySQL como gestor de base de datos, así el sistema integra funciones automatizadas para la creación de horarios, asignación de docentes y detección de conflictos, con una interfaz diseñada bajo principios de usabilidad, accesibilidad y coherencia visual institucional.

Implementación de la aplicación web

La implementación del sistema permitió reducir en promedio el 94,15% de los tiempos de ejecución de los procesos administrativos académicos. Se desarrollaron módulos para la gestión académica integral, permitiendo generar y editar horarios de forma automática y manual, detectar conflictos, publicar información y descargar reportes en diversos formatos.

Validación de la efectividad de la herramienta

La validación del sistema demostró mejoras significativas en la creación de horarios, asignación de docentes, resolución de conflictos y generación de reportes lo cual el sistema elimina solapamientos, garantiza la disponibilidad de recursos y mejora la experiencia de los usuarios, contribuyendo a la modernización de la gestión académica en la carrera.

6.2 Recomendaciones

A partir del desarrollo de la investigación y de los resultados obtenidos con la implementación de la aplicación web para la planificación académica de horarios en la carrera de Ingeniería en Software de la Universidad Laica Eloy Alfaro de Manabí. Ext El Carmen, se plantean las siguientes recomendaciones:

- Se recomienda al coordinador de la carrera adoptar de manera formal el uso del sistema desarrollado, integrándolo como una herramienta oficial para la elaboración y planificación de los horarios académicos, con el fin de reducir errores, optimizar tiempos y mejorar la organización administrativa en cada período académico.
- Es aconsejable capacitar al coordinador académico y a los docentes en el uso adecuado de la aplicación web, garantizando un aprovechamiento óptimo de todas sus funcionalidades, especialmente aquellas relacionadas con la generación automática de horarios, la edición manual y la adaptabilidad al sistema.
- Se sugiere mantener actualizada la base de datos del sistema, principalmente cuando se hagan posibles cambios en las mallas curriculares, asignaturas, docentes, aulas y períodos académicos, para asegurar la correcta generación de horarios y evitar inconsistencias en la creación de los horarios.
- Se recomienda implementar copias de seguridad periódicas de la base de datos, así como reforzar las medidas de seguridad del sistema, con el propósito de proteger la información de los datos y garantizar la disponibilidad del servicio ante posibles fallos técnicos o fuerza de la naturaleza.
- Es pertinente considerar a futuro la ampliación del sistema para que pueda ser utilizado por otras carreras de la institución, adaptando sus parámetros y configuraciones lo cual permitiría estandarizar la planificación académica y optimizar recursos a nivel institucional.
- Finalmente, se recomienda continuar con el mantenimiento y mejora continua del sistema, incorporando nuevas funcionalidades como notificaciones automatizadas, reportes estadísticos avanzados o integración con otros sistemas institucionales, con el fin de fortalecer la gestión académica y responder a las necesidades escalables de la institución.

Bibliografía

- Alvarado, M., & Quintero, J. (2021). *Gestión pedagógica universitaria: Innovación y planificación*. Editorial Académica del Conocimiento.
- Barragán, J. (2021). *Gestión académica universitaria: evolución y desafíos en la era digital*. Alfaomega Grupo Editor.
- Bernal, L., Helen, V., Pérez, O., & Renán, B. (2024). *Prototipo de aplicación web para la gestión del proceso de planificación académica de la universidad de guayaquil*. GUAYAQUIL – ECUADOR .
- Bonilla, D. (2020). *Requisitos de software: Análisis y validación*. Ra-Ma.
- Cabrera, M., & Arévalo, A. (2021). *Gestión del talento humano en instituciones educativas*. EduGestión. <https://doi.org/10.4578/edu.talento.cabrera2021>
- Camacho, F. (2023). *Planificación estratégica académica y gestión curricular*. Ediciones Macro.
- Castillo, C. (2020). *Transformación digital en la gestión universitaria*. Ediciones Académicas Españolas.
- Castro, R., & Montenegro, D. (2021). *Arquitecturas integradas para la gestión universitaria*. Editorial Universidad Central. <https://doi.org/10.9101/uce.castromont2021>
- Díaz, M. (2023). *Planeación y evaluación en la educación superior*. Editorial Trillas. <https://doi.org/10.1234/trillas.diaz2023>
- Ecler, C., & Luz, M. (2024). *Análisis Comparativo de Lenguajes de Programación: Python vs. Java en el Desarrollo de Aplicaciones Web* (Vol. 5). <https://doi.org/10.57166/micaela.v5.n1.2024.143>
- Escobar, F. (2021). *Diseño de sistemas informáticos*. Macro.

- Gamboa, A. J., Castillo, V. S., & Cano, C. A. (2024). *El proceso de investigación cualitativa: herramientas teórico-metodológicas para su desarrollo*. Editorial CUN.
- Gómez, J. A. (2021). *Aplicación web para la automatización y gestión de horarios de clases en la unidad educativa Juan XXIII*. Universidad Estatal Península de Santa Elena (Tesis de grado con ISBN y publicada como libro).
- González, C. P. (2023). *Diseño UX para aplicaciones educativas*. Ediciones UI/UX. <https://doi.org/10.5432/ux.design.education.perez2023>
- Guevara, M. Á. (2022). *Aplicaciones informáticas de tratamiento de texto. UF0320*. EDITORIAL TUTORFORMACION.
- Gutiérrez, P. (2023). *Implantación de sistemas de información*. Alfaomega.
- Hernández Sampieri, R. F., & Baptista Lucio, P. (2021). *Metodología de la investigación (7.ª ed.)*. McGraw-Hill.
- Hernández Sampieri, R., & Mendoza Torres, C. P. (2021). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta (7.ª ed.)*. McGraw-Hill Education.
- Hernández, J., & Rincón, A. (2023). *Pruebas de software: Estrategias para la calidad*. Marcombo.
- Hernández, M., & Silva, P. (2022). *Gestión de calidad en la educación superior: indicadores y evaluación*. Ediciones Calidad Educativa. <https://doi.org/10.5678/quality.he.diegue2022>
- Hernández, R., & López, J. (2023). *Transformación digital en la educación superior*. Editorial Innovación Académica. <https://doi.org/10.7654/transform.edusuper.hernandezlopez2023>
- Hurtado de Barrera, F. (2020). *Metodología de la investigación holística*. Episteme. <https://doi.org/ISBN 9789806370841>

- Jiménez, C., & Vargas, O. (2024). *Arquitectura de sistemas integrados para la educación superior*. Educación y Tecnología Press.
<https://doi.org/10.6543/integ.sysedu.jimenezvargas2024>
- López, C., & Carrasco, V. (2023). *Planificación académica inteligente: sistemas, validaciones y diseño adaptativo*. Editorial Universitaria Andina.
<https://doi.org/10.1234/eduandina.lopez2023>
- Martínez, L., & Ruiz, E. (2022). *Gestión de incidencias y cambio en sistemas educativos*. Editorial Gestión Académica. <https://doi.org/10.4567/incid.edu.martinezruiz2022>
- Méndez, A. (2022). *Historia de las tecnologías educativas*. Ediciones Pirámide.
- Mera, T. (2021). *Metodologías estructuradas para proyectos de grado en ingeniería*. Trillas.
- Morales, D. (2021). *Seguridad informática aplicada a sistemas educativos*. Ediciones CiberSeguridad Educativa. <https://doi.org/10.9871/seguridad.morales2021>
- Navarro, P., & López, M. (2023). *Trazabilidad y gobernanza en sistemas universitarios*. Editorial Gobernanza Educativa. <https://doi.org/10.9876/govern.edu.navarlop2023>
- Oca, P. Y. (2023). *Planificación, gestión y metodología en la Educación Superior*. Editorial Académica (posiblemente de Editorial Académica Nations, según índices).
<https://doi.org/10.56168/IBL.ED.167898>
- Oca, Y. M., Palau, N. R., & Sosa, E. T. (2023). *Planificación académica, gestión y metodología en la Educación Superior*. inBlue Editorial.
- Ocaña-Fernández, Y., & Luján-Mora, S. (2021). *Diseño de investigaciones educativas: Métodos cualitativos aplicados a entornos virtuales*. Editorial Octaedro.
<https://doi.org/9788419023089>.
- Olarte, L., & Rodríguez, D. (2022). *Metodologías de desarrollo de software*. Alfaomega.
- Ortega, A., & Suárez, D. (2022). *Simulación aplicada a la gestión universitaria*. Ediciones SimulAT. <https://doi.org/10.3210/simul.uni.ortegasuarez2022>

- Ortega, G., & Díaz, L. (2022). *Planeación estratégica universitaria: Enfoques y modelos*. Fondo Editorial Universitario. <https://doi.org/10.3210/planeacion.universitaria.ortegadiaz2022>
- Peña, L., & Rodríguez, M. (2022). *Gestión académica universitaria: teoría y práctica*. Editorial Alfa Educativa.
- Pérez, D., & Aguilar, K. (2022). *Capacitación tecnológica en universidades emergentes*. Ediciones Formación Plus. <https://doi.org/10.6789/fp.perezaguilar2022>
- Pressman, R. S., & Maxim, B. R. (2021). *Ingeniería del software*. McGraw-Hill.
- Pressman, R. S., & Maxim, B. R. (2021). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education. <https://doi.org/10.1036/0078022126>
- Ramírez, L. (2023). *Desarrollo web escalable: microservicios y personalización de sistemas*. Alfa Omega – TecnoEd. <https://doi.org/10.2345/alo.ramirez2023>
- Reinoso, L. (2020). *Modelos tradicionales en el desarrollo de software*. Ra-Ma.
- Rivera, J. (2021). *Desarrollo de aplicaciones web orientadas a procesos institucionales*. Alfaomega.
- Roldán, C. (2023). *Tecnologías educativas centradas en el estudiante*. Editorial Didáctica Digital. <https://doi.org/10.9932/educacion.tic.roldan2023>
- Ruben, B. D. (2022). *Implementing Sustainable Change in Higher Education: Principles and Practices of Collaborative Leadership*. Stylus Publishing.
- Salazar, E. (2022). *Indicadores de gestión educativa universitaria*. Editorial Universitaria de Gestión. <https://doi.org/10.4321/indicadores.salazar2022>
- Salazar, E. (2022). *Sistemas académicos personalizados: Diseño y adaptación institucional*. Universidad Técnica Particular de Loja.
- Sampieri, R. H., & Lucio, M. P. (2022). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill. https://doi.org/ISBN_9781456264024

- Sampieri, R. H., & Lucio, P. B. (2022). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill Education. <https://doi.org/10.1016/C2019-0-03120-1>
- Sarango, A. F., Pallmay, E. R., Sarzosa, J. P., & Pozo, J. E. (2024). *Tipos y clasificación de las investigaciones*. LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades, Asunción, Paraguay. <https://doi.org/10.56712/latam.v5i2.1927>
- Schach, S. R. (2022). *Software Engineering*. McGraw-Hill.
- Sierra Bravo, R. (2021). *Técnicas de investigación social: Teoría y ejercicios prácticos*. Paraninfo. [https://doi.org/ ISBN9788428340934](https://doi.org/ISBN9788428340934)
- Smith, M. D. (2023). *The Abundant University: Remaking Higher Education for a Digital World*. MIT Press.
- Sommerville, I. (2020). *Ingeniería del software*. Pearson Educación.
- Tapia Camargo, M. S. (2020). *Un enfoque cuantitativo de la investigación*. Universidad Nacional de Huancavelica (Repositorio institucional).
- Zambrano, D., & Vargas, K. (2024). *Frontend moderno: HTML5, CSS3 y frameworks interactivos*. Editorial WebTech.

Anexos

Anexo A. Aprobación de tema



Universidad Laica Eloy Alfaro de Manabí

Periodo 2025-1 - Notificación de tutor asignado - TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Estimad@
Docente y Estudiante
Uleam

En cumplimiento de lo establecido en la Ley, el Reglamento de Régimen Académico y las disposiciones estatutarias de la Uleam, por medio de la presente se oficializa la dirección y tutoría en el desarrollo del Trabajo de Integración curricular / Trabajo de Titulación del siguiente estudiante:

Tema: APLICACIÓN WEB PARA LA PLANIFICACIÓN ACADÉMICA DE HORARIOS EN LA CARRERA DE INGENIERÍA EN SOFTWARE EN ULEAM EXTENSIÓN EL CARMEN.

Estado de aprobación: Aprobado

Tipo de titulación: Trabajo de Integración Curricular

Tipo de proyecto: Trabajo de Integración Curricular / Trabajo de titulación se articula con proyectos y programas de Investigación.

Apellidos y nombres del tutor asignado: MENDOZA VILLAMAR ROCIO ALEXANDRA

Apellidos y nombres del estudiante: FUERES TIPANTUÑA WILMER HERNAN

Carrera: TECNOLOGÍAS DE LA INFORMACIÓN 2022 (EL CARMEN)

Periodo de inducción: Periodo 2025-1

Sírvase cumplir con lo dispuesto en el Manual de Procedimientos de TITULACIÓN DE ESTUDIANTES DE GRADO: TRABAJO DE INTEGRACIÓN CURRICULAR Y UNIDAD DE TITULACIÓN <https://departamentos.uleam.edu.ec/gestion-aseguramiento-calidad/files/2020/06/PAT-04-Titulacion-de-Estudiantes-de-grado-UIC-y-UT.pdf>

Particular que se informa para los fines consiguientes.

Atentamente,

Comisión Académica y Responsable de Titulación.

Anexo B. Instrumento entrevista



UNIVERSIDAD LAICA "ELOY ALFARO" DE MANABÍ
EXTENSIÓN EN EL CARMEN
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Creada Ley No.10 - Registro Oficial 313 de noviembre de 13 de 1985

Entrevista para obtener información que servirá para el proceso de mi titulación

Tema: Aplicación web para la planificación académica de horarios en la carrera de Ingeniería en Software en la ULEAM Extensión El Carmen

Objetivo: Obtener información detallada sobre el proceso actual de planificación académica, herramientas utilizadas y percepciones del coordinador sobre la posible implementación de una aplicación web.

ENTREVISTA DIRIGIDA AL COORDINADOR DE CARRERA

1. ¿Cómo describe el proceso actual de planificación académica de horarios en la carrera de Ingeniería en Software?
2. ¿Qué expectativas tiene sobre el impacto de una aplicación web en el contexto académico actual de la carrera?
3. ¿Considera que el proceso actual de elaboración de horarios es eficiente?
4. ¿Qué nivel de dificultad considera que tiene actualmente el proceso de planificación académica de horarios?
5. ¿Ha experimentado conflictos de horarios (choques de materias, aulas, u horarios repetidos)?
6. ¿Con qué frecuencia los horarios planificados son modificados durante el semestre?
7. ¿Considera necesario implementar una aplicación web para este proceso? ¿Por qué?
8. ¿Qué funcionalidades considera esenciales que debería tener una herramienta informática para este propósito?
9. ¿Qué beneficios considera que podría aportar una aplicación web al proceso de planificación académica?
10. ¿Qué aspectos considera importantes para lograr que el personal docente y administrativo adopte y utilice la aplicación?

Anexo C. Instrumento encuesta

Revisión: Encuesta Dirigida a los Docentes de la Carrera de Software de la ULEAM-Extensión El Carmen

Contactos Preguntas



1. ¿Con qué frecuencia participa usted

0.0/0

Puntos

1. ¿Con qué frecuencia participa usted en la elaboración de horarios académicos? *

Punt

- Siempre
- Algunas veces
- Rara vez
- Nunca

2. ¿Quiénes realizan los horarios? *

- Cordinador
- Docentes
- Coordinador y docente
- Comisión académica

3. ¿Considera que el proceso actual de elaboración de horarios es eficiente? *

- Sí
- Parcialmente
- No

4. ¿Qué nivel de dificultad considera que tiene actualmente el proceso de planificación académica de horarios? *

- Bajo
- Moderado
- Alto

5. ¿Ha experimentado conflictos de horarios (choques de materias, aulas, u horarios repetidos)? *

- Sí
- No

6. ¿Con qué frecuencia los horarios planificados son modificados durante el semestre? *

- Nunca
- 1 a 2 veces
- Más de 2 veces

7. ¿Cuenta con acceso a un sistema o plataforma institucional para visualizar su horario? *

- Sí
- No

8. ¿Está de acuerdo con que una herramienta tecnológica podría facilitar el proceso de planificación y organización de los horarios académicos de clases en la carrera de Ingeniería en Software? *

- Sí
- No
- No estoy de acuerdo

9. ¿Qué tiempo promedio se tardan actualmente en la elaboración de horarios? *

- Menos de 1 día
- De 1 a 3 días
- De 4 a 7 días
- De 1 a 2 semanas
- Más de 2 semanas

10. ¿Qué tiempo promedio se tardan en realizar un cambio por choques o disponibilidad de horas al momento de la elaboración de los horarios? *

- Menos de 1 hora
- Entre 1 y 3 horas
- Entre 4 y 8 horas
- Entre 1 y 2 días
- Más de 2 días

11. ¿Estaría dispuesto a utilizar una aplicación web para elaborar y consultar los horarios asignados? *

- Sí
- No
- Tal vez

12. ¿Qué funcionalidades considera más útiles en una aplicación para planificación de horarios? (Seleccione 2 o más)

*


- Visualización del horario
- Notificaciones de cambio
- Acceso desde celular
- Generación de reportes
- Consulta de aulas asignadas

Anexo D. Entrevista al coordinador



Descripción: Se realizó la encuesta dirigida al coordinador de la carrera de software de manera presencial para obtener una información precisa.

Anexo E. Certificado de coincidencia académica (emitido por tutor del sistema antiplagio).



CERTIFICADO DE ANÁLISIS
magister

TESIS_WILMER FUERES

6%
Textos sospechosos

- < 1% Similitudes
- 0 % similitudes entre comillas
- 0 % entre las fuentes mencionadas
- 1% Idiomas no reconocidos
- 4% Textos potencialmente generados por la IA


Nombre del documento: TESIS_WILMER FUERES.docx
ID del documento: bdb51bab1a8a325d734cbf4c0f981f9e2d9059ac
Tamaño del documento original: 5,72 MB

Depositante: ROCIO MENDOZA VILLAMAR
Fecha de depósito: 29/1/2026
Tipo de carga: interfaz
fecha de fin de análisis: 29/1/2026






Número de palabras: 21.347
Número de caracteres: 145.566


Ubicación de las similitudes en el documento:

Fuente principal detectada

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 Evelyn_Velazquez_10 - copia.docx Evelyn_Velazquez_10 - copia #1b90e8 <small>Viene de de mi grupo</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (15 palabras)</small>

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	 TESIS APQC 20_7_2023 correccion(1).docx TESIS APQC 20_7_2023 correc... #088cfb <small>Viene de de mi grupo</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (20 palabras)</small>
2	 IMPACTO DE LA INTELIGENCIA ARTIFICIAL.pdf IMPACTO DE LA INTELI... #c78e7b <small>Viene de de mi grupo</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (13 palabras)</small>
3	 vaciadosmadrid.es La metodología en cascada: fases y características <small>https://vaciadosmadrid.es/blog/que-es-metodologia-en-cascada-waterfall-fases/</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (15 palabras)</small>
4	 Documento de otro usuario #1e93bd <small>Viene de de otro grupo</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (10 palabras)</small>
5	 TESIS FINAL YM-jT.pdf Tesis Titulación #72d9bc <small>Viene de de mi grupo</small>	< 1%	<div style="width: 100%; height: 10px; background-color: #0070c0;"></div>	<small>Palabras idénticas: < 1% (10 palabras)</small>



111

Glosario

- **Aplicación web:**

Sistema de software que se ejecuta en un navegador web y permite a los usuarios interactuar con información y funcionalidades a través de internet o una red local. En el presente proyecto, la aplicación web permite la planificación, gestión y visualización de horarios académicos.

- **API REST:**

Interfaz de programación de aplicaciones basada en el protocolo HTTP que permite la comunicación entre el frontend y el backend del sistema, facilitando el intercambio de datos de forma segura y estructurada.

- **Arquitectura de tres capas:**

Modelo de diseño de software que divide el sistema en tres niveles principales: presentación (frontend), lógica de negocio (backend) y base de datos, mejorando la escalabilidad, mantenimiento y seguridad del sistema.

- **Autenticación:**

Proceso mediante el cual el sistema verifica la identidad de un usuario a través de credenciales como nombre de usuario y contraseña, permitiendo el acceso únicamente a usuarios autorizados.

- **Aula:**

Espacio físico destinado al desarrollo de actividades académicas, el cual es considerado como un recurso dentro del sistema para la correcta asignación de horarios.

- **Base de datos:**

Conjunto estructurado de datos almacenados electrónicamente que permite guardar, consultar y administrar información de manera eficiente. En el sistema desarrollado se utiliza MySQL como gestor de base de datos.

- **Backend:**

Parte del sistema encargada de procesar la lógica de negocio, gestionar la base de datos y responder a las solicitudes del frontend. En este proyecto fue desarrollado utilizando PHP.

- **Carga académica:**

Conjunto de horas y asignaturas asignadas a un docente durante un período académico determinado.

- **Carrera:**

Programa académico ofertado por la institución, como Ingeniería en Software o Tecnologías de la Información, utilizado en el sistema para organizar los horarios académicos.

- **Control de acceso:**

Mecanismo de seguridad que limita las acciones que un usuario puede realizar en el sistema según su rol, garantizando la correcta administración de permisos.

- **Coordinador académico:**

Usuario con privilegios administrativos dentro del sistema, responsable de gestionar usuarios, registrar información académica y generar horarios.

- **CRUD:**

Acrónimo de Create, Read, Update y Delete, operaciones básicas para la gestión de datos dentro de un sistema de información.

- **Docente:**

Usuario del sistema encargado de impartir asignaturas y consultar los horarios académicos que le han sido asignados.

- **Exportación de datos:**

Funcionalidad que permite descargar la información del sistema en diferentes formatos como PDF, Excel, CSV o HTML.

- **Frontend:**

Parte visual del sistema con la que interactúa el usuario, desarrollada utilizando React, encargada de mostrar la información de forma amigable e intuitiva.

- **Gestión académica:**

Proceso administrativo que involucra la organización de materias, docentes, aulas, niveles, paralelos y jornadas dentro de una institución educativa.

- **Gestión de usuarios:**

Funcionalidad del sistema que permite crear, modificar, eliminar y administrar cuentas de usuario con diferentes roles.

- **Horario académico:**

Distribución organizada de asignaturas, docentes, aulas y horarios dentro de un período académico.

- **Interfaz de usuario:**

Diseño visual que permite la interacción entre el usuario y el sistema, enfocada en la facilidad de uso y accesibilidad.

- **Jornada:**

Periodo de tiempo en el cual se desarrollan las actividades académicas, como jornada matutina o vespertina.

- **Malla curricular:**

Estructura académica que define las asignaturas que deben cursarse en cada nivel de una carrera universitaria.

- **Metodología en cascada:**

Modelo de desarrollo de software secuencial que divide el proyecto en fases claramente definidas como análisis, diseño, implementación, pruebas y mantenimiento.

- **Nivel académico:**

Etapas o semestres dentro de una carrera universitaria, utilizado en el sistema para organizar las materias y horarios.

- **Paralelo:**

Grupo académico que corresponde a un mismo nivel y carrera, permitiendo la organización de estudiantes en diferentes secciones.