



UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ

Facultad de Ciencias de la Vida y Tecnología;

Carrera de Ingeniería en Tecnologías de la Información.

Proyecto Integrador Previo A La Obtención De Título De:

Ingeniero en Tecnologías de la información.

T:

Desarrollo de un Sistema para la Gestión y Administración del Uso de Escenarios en la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta.

Autores:

Castillo Vera Ariel Alexander & Cedeño Mosquera Jostin Abraham

Tutor:

Ing. Junior Zamora Mendoza

Manta - Ecuador

2025(2)

DECLARACIÓN DE AUTORÍA.

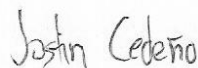
Nosotros, **Castillo Vera Ariel Alexander** y **Cedeño Mosquera Jostin Abraham**, declaramos bajo juramento que el trabajo de titulación titulado: *“Desarrollo de un Sistema para la Gestión y Administración del Uso de Escenarios en la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta”*, es de nuestra autoría y responsabilidad académica.

Afirmamos que este documento es completamente original y no ha sido presentado total ni parcialmente para la obtención de ningún otro título o grado académico. Todas las ideas, fuentes de información, bibliografía y herramientas tecnológicas utilizadas durante la investigación y desarrollo del software han sido debidamente reconocidas y citadas siguiendo los lineamientos de las normas APA 7ma edición.



Castillo Vera Ariel Alexander

C.I: 1350141113



Cedeño Mosquera Jostin Abraham

C.I: 1350797955

CERTIFICACIÓN DEL TUTOR.

	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 2

CERTIFICACIÓN

En calidad de docente tutor de la Facultad de Ciencias de la Vida y Tecnologías de la sede Manta 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 **CASTILLO VERA ARIEL ALEXANDER**, legalmente matriculado en la carrera de Tecnologías de la Información, período académico 2005-2026, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"DESARROLLO DE UN SISTEMA PARA LA GESTIÓN Y ADMINISTRACIÓN DEL USO DE ESCENARIOS EN LA UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ, MATRIZ MANTA"**.

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.

Manta, 3 de febrero de 2026.

Lo certifico,


Ing. Junior José Zamora Mendoza, Mg.
Docente Tutor
Área: Desarrollo de Software

	NOMBRE DEL DOCUMENTO:	CÓDIGO: PAT-04-F-004
	CERTIFICADO DE TUTOR(A).	
	PROCEDIMIENTO: TITULACIÓN DE ESTUDIANTES DE GRADO BAJO LA UNIDAD DE INTEGRACIÓN CURRICULAR	REVISIÓN: 1 Página 2 de 2

CERTIFICACIÓN

En calidad de docente tutor de la Facultad de Ciencias de la Vida y Tecnologías de la sede Manta de la Universidad Laica "Eloy Alfaro" de Manabí, CERTIFICO:

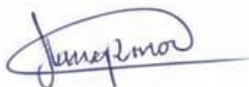
Haber dirigido, revisado y aprobado preliminarmente el Trabajo de integración Curricular bajo la autoría del estudiante **CEDEÑO MOSQUERA JOSTIN ABRAHAM**, legamente matriculado en la carrera de Tecnologías de la información, periodo académico 2005-2026, cumpliendo el total de 384 horas, cuyo tema del proyecto es **"DESARROLLO DE UN SISTEMA PARA LA GESTIÓN Y ADMINISTRACIÓN DEL USO DE ESCENARIOS EN LA UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ, MATRIZ MANTA"**.

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 cometida 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.

Manta, 3 de febrero de 2026.

Lo certifico.



Ing. Junior José Zamora Mendoza, Mg.
Docente Tutor
Área: Desarrollo de Software

DEDICATORIA.

Dedico este trabajo de tesis con profunda gratitud hacía mi familia, en especial a mis padres, Narcisa Mirella Vera y Jonny Castillo Gorozabel, que sin su apoyo y sacrificio incondicional no hubiera podido llegar a dónde estoy ahora, y he podido cumplir todas y cada una de mis metas hasta el momento.

A todo quién estuviera apoyándome en esta etapa de mi vida para poder superar los obstáculos de la misma, a la música que me ha acompañado en toda esta etapa desde que era pequeño. Este logro es por y para ustedes.

Gracias.

Castillo Vera Ariel Alexander.

AGRADECIMIENTOS.

Les expresamos nuestro más sincero y profundo agradecimiento a los ingenieros Junior Zamora Mendoza y Óscar Gonzalez, quienes, a lo largo de todo el desarrollo de esta tesis, nos ayudaron y brindaron su valioso conocimiento académico y técnico. Tener la disposición tanto su experiencia y apoyo fueron fundamentales para orientar adecuadamente el proceso tanto de investigación como de desarrollo y fortalecer cada una de las etapas del trabajo realizado.

Agradecemos de manera especial sus acertadas observaciones y la gran experiencia que llevan consigo, recomendaciones oportunas y estar acompañándonos constantemente para poder lograr esta etapa de la mejor manera, mejorar tanto cómo el código del desarrollo y la investigación para poder garantizar el cumplimiento de los objetivos planteados. Su guía y criterio profesional permitieron superar diversas dificultades presentadas durante el desarrollo de la tesis, con éxito, haciendo que el proyecto tenga más solidez y claridad.

Por último, valoramos y reconocemos el tiempo dedicad, la paciencia y la confianza dedicada en nosotros dos durante todo este proceso académico, los cuales fueron un importante estímulo para poder culminar satisfactoriamente esta investigación.

Índice

DECLARACIÓN DE AUTORÍA.....	2
CERTIFICACIÓN DEL TUTOR.....	3
DEDICATORIA.	5
AGRADECIMIENTOS.	6
Resumen.....	13
Abstracst	14
Capítulo I: El Problema	15
1.1. Problema	15
1.1.1 Planteamiento del problema.	15
1.1.2 Formulación del problema.	16
1.2. Objetivos	16
1.1.3 Objetivo General	16
1.1.4 Objetivos Específicos	16
1.3. Justificación.....	17
1.4. Diagrama Causa-Efecto	18
Capítulo II: Marco Teórico	19
2.1. Antecedentes.	19
2.1.1 Sistemas de Gestión de Espacios Universitarios en el Contexto Global.	19
2.1.2 Transformación digital.....	20
2.1.3 Gestión de recursos físicos.	20
2.1.4 Sistemas de información.....	20
2.1.5 Aplicaciones web.....	21
2.1.6 Tecnologías y arquitectura a emplear.	22
2.2. Categorías Conceptuales.....	25

2.2.1	ReactJS.....	25
2.2.2	Axios.	25
2.2.3	Zustand.....	25
2.2.4	TailwindCSS.	25
2.2.5	Laravel.....	25
2.2.6	PHP.	26
2.2.7	Eloquent ORM.....	26
2.2.8	PostgreSQL	26
2.2.9	Clean Architecture.....	26
2.2.10	Arquitectura Monolítica.	26
2.3.	Trazabilidad.....	27
2.4.	Sistema de reserva.....	27
2.5.	Marco Legal.....	27
2.5.1	Introducción	27
2.5.2	Bases Constitucionales.....	28
2.5.3	Legislación Nacional.	28
2.6.	Conclusiones del marco teórico.	29
Capítulo III: Metodología		30
3.1.	Introducción	30
3.2.	Catálogo de partes interesadas y actores.....	30
3.3.	Recolección de Datos.....	31
3.4.	Población y Muestra.	31
3.4.1	Fórmula aplicada.....	31
3.5.	Métodos de investigación.	32
3.5.1	Métodos Teóricos.....	32

3.5.2	Métodos Empíricos.	32
3.6.	Técnicas e Instrumentos.....	32
3.6.1	Técnica: Encuesta estructurada.....	32
3.6.2	Técnica: Entrevista semiestructurada.	32
3.7.	Levantamiento de Procesos del Negocio (As-Is).....	33
3.8.	Descripción de la Metodología de Desarrollo y Selección de Artefactos ...	33
3.8.1	Justificación de la Metodología (XP)	33
3.8.2	Selección de Artefactos y Prácticas de XP	34
Capítulo IV: Propuesta o Desarrollo del Proyecto.....		35
4.1.	Introducción	35
4.2.	Análisis del Diagnóstico	36
4.2.1	Evaluación de la gestión actual.....	36
4.3.	Metodología de desarrollo seleccionada.....	45
4.4.	Organización de Iteraciones y Roles.....	46
4.5.	Cronograma de Desarrollo Detallada.....	50
4.6.	Diseño de la base de datos.	52
4.7.	Descripción Detallada de las Entidades Principales y sus Relaciones	54
4.8.	Arquitectura del Sistema.....	55
4.8.1	Componentes Clave de la Arquitectura	56
4.9.	Desarrollo del Backend.....	56
4.10.	Desarrollo del Frontend	71
4.11.	Conclusiones del Capítulo	76
Capítulo V: Evaluación de Resultados		77
5.2.	Descripción del proceso de evaluación.....	77
5.2.1	Criterios de Evaluación.....	77

5.2.2	Relación con los objetivos del proyecto	79
5.3.	Presentación de resultados	79
5.3.1	Resultados técnicos y funcionalidades implementadas	79
5.3.2	Evidencias de pruebas realizadas.....	83
5.4.	Análisis e interpretación de resultados	91
5.5.	Entrega del producto final.....	92
	Conclusiones	95
	Recomendaciones	96
	Anexos	98
	Anexo 1: Cuestionario para estudiantes, docentes, personales administrativos y usuarios externos	98
	Anexo 2: Enlace al instrumento de recolección de datos (Formulario Digital)	100
	Anexo 3: Acceso al Diagrama de Entidad-Relación interactivo.....	101
	Anexo 4: Tutoría Guía con Docente Tutor y Personal Administrativo	102
	Bibliografía	103

Índice de tablas

<i>Tabla 1. Tecnologías usadas en el Frontend.</i>	22
<i>Tabla 2. Tecnologías usadas en el Backend.</i>	24
Tabla 3. Historia de Usuarios y Criterios de aceptación.....	47
<i>Tabla 4. Cronograma de actividades</i>	50
Tabla 5. Endpoints utilizados.....	59
Tabla 6. Resultados parciales.....	78
Tabla 7. Matriz de pruebas.....	79

Índice de figuras.

Figura 1. Diagrama Causa-Efecto.....	18
Figura 2. Cuál es la percepción de la gestión actual.	36
Figura 3. Es fácil entender el proceso actual.	37
Figura 4. Gráfico sobre si los pasos para reservar un espacio son claros.	38
Figura 5. Gráfico sobre si hay problemas o demoras al querer reservar un espacio....	39
Figura 6. Preferencia sobre existencia de una plataforma digital.	40
Figura 7. Requerimientos de un sistema web.	41
Figura 8. Los usuarios se sentirían seguros con historial y seguimiento.	42
Figura 9. Mejoraría la transparencia en la asignación de espacios con un sistema web.	43
Figura 10. Usuarios recomendarían la plataforma web.	44
Figura 11. Iteraciones Dentro del Backend.....	46
Figura 12. Diagrama Entidad-Relación de la base de datos del sistema.	53
Figura 13. Arquitectura utilizada en el desarrollo del sistema.....	55
Figura 14. Método encargado del manejo y validación de fechas.	57
Figura 15. Prueba de endpoint de listado de reservas con Postman.	58
Figura 16. Formulario para el inicio de sesión.	71
Figura 17. Panel de vista de los espacios con los que cuenta la institución.	72
Figura 18. Vista de un espacio detallado con precio, mapa y equipamiento.	73
Figura 19. Dashboard para la administración de todos los espacios.....	74
Figura 20. Formulario de crear nuevo espacio.....	74
Figura 21. Dashboard para la gestión de usuarios.	75

Figura 22. Dashboard para la gestión de roles.	75
Figura 23. Body Json para hacer el POST de espacios.....	83
Figura 24. Respuesta correcta del POST de espacios.	84
Figura 25. Devolución de la respuesta usando GET en espacios.	85
Figura 26. Body JSON para el post de usuarios.	86
Figura 27. Respuesta correcta del POST usuarios.	86
Figura 28. Respuesta devuelta de todos los usuarios.	87
Figura 29. Body JSON para el POST de roles.....	87
Figura 30. Respuesta correcta del POST de roles.....	88
Figura 31. GET de todos los roles.	88
Figura 32. Pantalla de Login.....	89
Figura 33. Pantalla Home	89
Figura 34. Pantalla Dashboard Administrativo.....	90
Figura 35. Interfaz de Formulario.....	90
Figura 36. Token Personal	91
Figura 37. Reservación de espacios.	92
Figura 38. Reservación de espacios 2	93
Figura 39. Reservación de espacios 3.....	93
Figura 40. Revisión de solicitudes.....	94
Figura 41. Información general de las solicitudes.	94

Resumen

Título: *Desarrollo de un Sistema para la Gestión y Administración del Uso de Escenarios en la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta.*

Antecedentes: La presente tesis tiene como objetivo el desarrollo de una aplicación web destinada a la gestión de escenarios universitarios en la Universidad Laica Eloy Alfaro de Manabí (ULEAM), de tal manera que podamos mejorar aquellos procesos de solicitud, administración y control de los espacios de uso común para eventos de diferentes categorías. El problema que llegamos a encontrar dentro del proyecto de tesis es en la total falta de un sistema web que nos permita poder gestionar de manera eficiente y segura todos los espacios de la universidad lo cual genera distintos problemas al momento de querer utilizar cualquier espacio, como retrasos o dificultades de disponibilidad de los mismos.

Para la solución que proponemos se implementó una aplicación web basada en una arquitectura de backend desacoplada, utilizando como tecnologías el framework Laravel como núcleo del sistema, junto a una base de datos relacional en PostgreSQL, además de un cliente en React, Este sistema expone los servicios mediante una API REST, permitiendo que se pueda interactuar con diferentes clientes y facilitando la escalabilidad y el mantenimiento de la aplicación. Asimismo, en todo el proyecto se utilizó buenas prácticas de desarrollo de software y clean code, control de accesos y validación de datos.

Así se consiguió un sistema web que mejoraría sustancialmente la gestión de espacios de la ULEAM, quitar todo el proceso manual que ha menudo tienen problemas por algo que mejora muchísimo cualquier aspecto de este mismo, demostrando la viabilidad y utilidad de nuestra propuesta de tesis.

Abstrasct

Title: *Development of a System for the Management and Administration of the Use of Scenarios at the Universidad Laica Eloy Alfaro de Manabí, Matriz Manta.*

Background: The objective of this thesis is to develop a web application for managing university venues at the Universidad Laica Eloy Alfaro de Manabí (ULEAM), in order to improve the processes of requesting, administering, and controlling common spaces for events of different categories. The problem identified centers on the total lack of a centralized digital system that allows for efficient, secure and organized management of these venues, which leads to delays, duplication of information, and difficulties in controlling availability.

For the solution we propose, a web application based on a decoupled backend architecture was implemented, using the Laravel framework as the core of the system, together with a relational database in PostgreSQL, as well as a client in React. This system exposes the services through a REST API, allowing interaction with different clients and facilitating the scalability and maintenance of the application. Likewise, throughout the project, we use good software development practices and clean code, access control, and data validation.

Thus, a web system was achieved that would substantially improve the management of ULEAM spaces eliminating the entire manual process that has often had problems due to something that greatly improves any aspect of it, demonstrating the viability and usefulness of our thesis proposal.

Capítulo I: El Problema

1.1. Problema

1.1.1 Planteamiento del problema.

La gestión de las áreas de la Universidad Laica Eloy Alfaro de Manabí, matriz Manta como escenarios, chancas, auditorios entre otros, son un eje importante para cualquier institución educativa superior y saber gestionarla y manejarla es fundamental para una óptima gestión. Según (UNESCO, 2022), la innovación digital es una de las mejores herramientas de cualquier institución educativa, enriqueciéndola, mejorando su gestión y gobernanza. Aun así, a nivel global y nacional, algunas instituciones educativas como las universidades presentan muchos casos sobre una mala administración de sus bienes físicos, lo cual limita su capacidad operativa y administrativa.

De esta manera, la Universidad Laica Eloy Alfaro de Manabí, tiene una amplia variedad de escenarios físicos, aquellos son utilizados diariamente por estudiantes, docentes, personal administrativo y usuarios externos. Aunque, la reservación, control y mantenimiento de dichos espacios se realiza de manera manual, sin un sistema con el que se pueda gestionar las solicitudes de manera eficiente. Ya que en Ecuador la secretaria nacional de Educación Superior, Ciencia, Tecnología e Innovación establece estrategias para que la gestión institucional se realice mediante el uso de tecnologías de la información. (Senecyt, 2022).

En el contexto de esta tesis, el término “escenarios universitarios” engloba toda la infraestructura física disponible dentro de la institución para el desarrollo de actividades académicas, culturales, deportivas y administrativas. Esto hace referencia de manera específica a: aulas de clases, auditorios, canchas deportivas, plazas abiertas a procesos de reserva y asignación institucional.

Esta situación no es conveniente para la universidad: duplicación de reservas, falta de trazabilidad, sin disponibilidad en tiempo real y dificultad al momento de la planificación institucional. Aparte, los usuarios deben realizar el proceso de solicitud de uso del espacio de forma presencial, lo que representa una pérdida de tiempo tanto para ellos como para el personal administrativo encargado. Afectando no solo al control interno sino a toda la

institución esta falta de un control, sino que también representa una brecha frente a las exigencias tecnológicas actuales.

En esta situación, identificamos la gran necesidad de desarrollar un sistema web para la gestión de los espacios de la universidad, que nos permita hacer todo el proceso desde un solo lugar, completamente administrable y disponible siempre, tanto para usuarios internos de la universidad como estudiante, docentes, personal administrativo y usuarios externos, garantizando trazabilidad, transparencia y eficiencia administrativa. Implementar este sistema web ayudará significativamente a ir moviéndonos hacia una modernización administrativa y gestión de recursos disponibles.

1.1.2 Formulación del problema.

¿Cómo puede el desarrollo de un sistema web mejorar la gestión, control y trazabilidad de los espacios universitarios en la Universidad Laica Eloy Alfaro de Manabí para satisfacer el requerimiento de actualizar los procedimientos manuales que se usan hoy en día para administrar y reservar?

1.2. Objetivos

1.1.3 Objetivo General

Desarrollar un sistema web para la administración y gestión de los escenarios de la universidad, aplicando tecnologías de la información, para poder optimizar el uso de estas y facilitar el proceso de solicitud, teniendo de esta manera una gestión más eficiente y transparente de estos recursos institucionales.

1.1.4 Objetivos Específicos

- Desarrollar un sistema web que nos permita gestionar los espacios de la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta y generar solicitudes de uso, empleando tecnologías actuales para optimizar la eficacia y transparencia del proceso.
- Implementar en el sistema web un formulario de registro y autenticación que permita a todos los usuarios tanto internos como externos solicitar el uso de algún espacio, garantizando accesibilidad, control y seguridad en el proceso.
- Construir la funcionalidad de alquiler de escenarios, que permita registrar solicitudes por parte de los usuarios y ser revisada para su aprobación o negación y permita la gestión eficiente de estas mismas.

1.3. Justificación.

Según (Morillo, 2018) poder implementar un sistema web automatizado de gestión administrativa permite poder optimizar los procesos internos, mejorar la eficiencia operativa y garantizar una adecuada administración de los recursos institucionales. Al poder estar haciendo el proceso de manera digital y poder mejorar estos procesos universitarios, se vuelve indispensable la implementación de un sistema que nos permita todo eso. Además, la correcta administración de estos espacios garantiza una planificación más estratégica de las actividades universitarias, evitando superposiciones y promoviendo un uso equitativo de los recursos.

De igual manera, necesitamos tener en cuenta que el adecuado manejo de la información que se usa para los escenarios universitarios es clave para la transparencia institucional y la toma de decisiones. Según (Aranda, 2013) si los documentos no se organizan y se mantienen adecuadamente de modo que puedan ser recuperados con facilidad en cualquier momento, no importa lo que esté pasando, no tendrán ninguna función útil, lo cual afecta directamente la rendición de cuentas y el cumplimiento de los objetivos que requiere la universidad cada año. En el caso de la Universidad Laica Eloy Alfaro de Manabí, la ausencia de no tener registros de manera digitalizada y guardada podría indicar problemas al momento de tener auditorías hacia la universidad.

A ello hay que tener en cuenta que las instituciones modernas deben orientarse hacia una gestión basada en el control y el monitoreo constante del desempeño de sus procesos, de esa manera se puede ser mucho más eficiente con estos mismos procesos de los que se habla. De esta manera, tanto el diseño como la implementación de un sistema web para administrar y gestionar las solicitudes de los espacios universitarios ofrece una solución que necesita la Universidad Eloy Alfaro de Manabí. Kework dice que el diseño de un sistema de control de gestión ayuda a fortalecer el seguimiento y logro de resultados, dándonos la posibilidad de medir los avances del área en buscar de una mejora en la calidad de la formación. Por eso, el sistema web que proponemos en esta tesis pretende no solo mejorar la administración operativa, sino también convertirse en una herramienta de apoyo a la mejora continua de la universidad. (Kework, 2019).

1.4. Diagrama Causa-Efecto

Figura 1. Diagrama Causa-Efecto

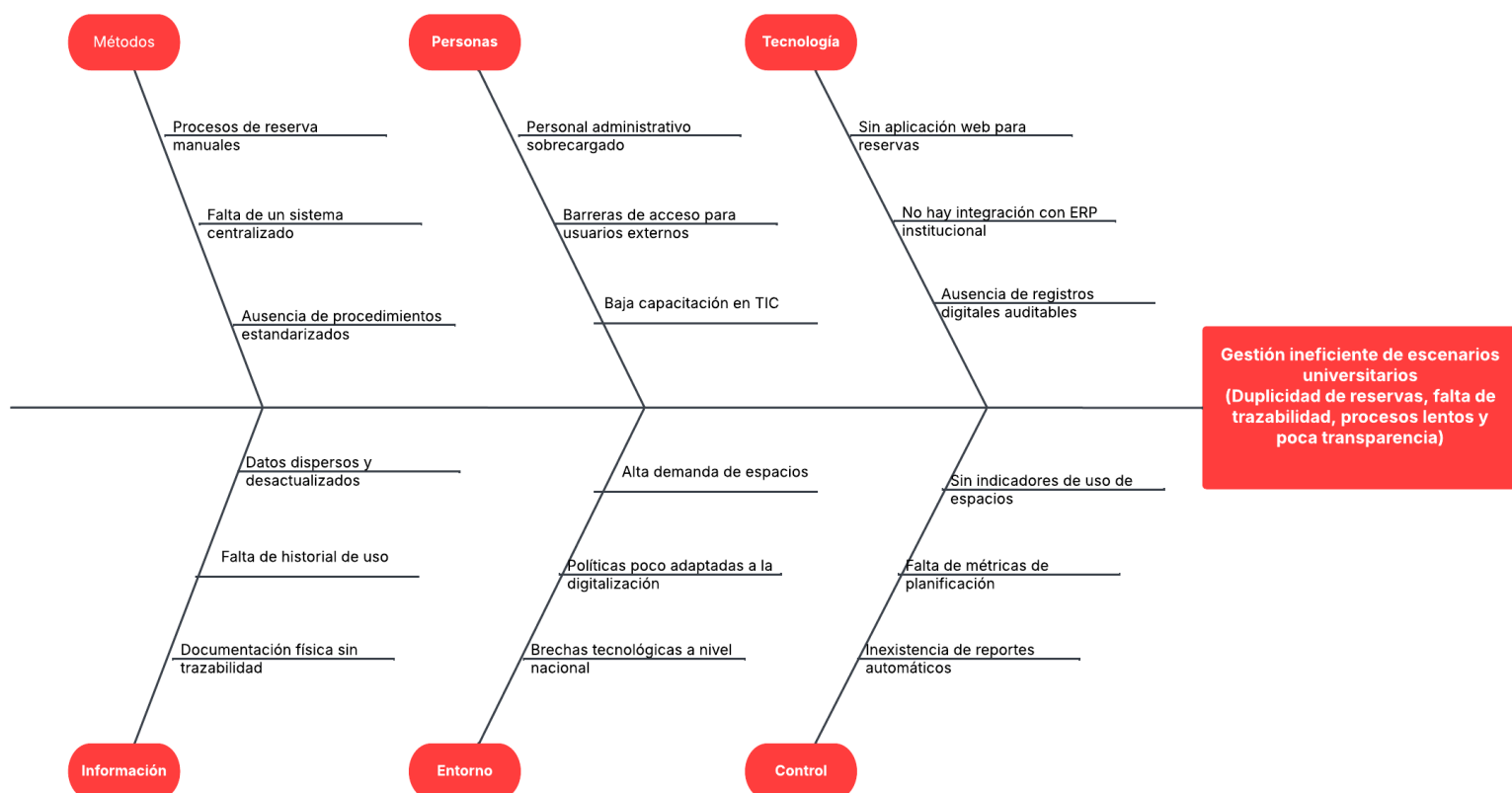


Diagrama de causa-efecto sobre la gestión de escenarios universitarios en la ULEAM.

Capítulo II: Marco Teórico

2.1. Antecedentes.

2.1.1 Sistemas de Gestión de Espacios Universitarios en el Contexto Global.

(Flores-Xolocotzi, 2022) comentan que el 78% de las universidades en países de primer mundo han implementado sistemas digitales para administrar estas aulas, auditorios y áreas deportivas, reduciendo problemas de reservas manuales en un 40% mediante algoritmos de asignación inteligente. Podemos destacar un caso emblemático en el Instituto Tecnológico de Massachusetts (MIT), donde agregar sensores IoT y utilizar un sistema de reservas en tiempo real mejoró la utilización de espacios en un 92%.

Aquella herramienta fue diseñada para automatizar procesos clave como la aplicación de exámenes, la asignación de calificaciones y la generación de informes académicos, eliminando redundancias y aumentando la accesibilidad de la información. Según (Doctor A. C., 2022) este sistema permitió “evitar la redundancia de datos y esfuerzos, y proporcionar información accesible y confiable sobre exámenes y calificaciones”.

Por otra parte, en el contexto ecuatoriano dice (UPS, 2024) diversas universidades también han comenzado procesos de transformación digital con el objetivo de modernizar su infraestructura tecnológica y mejorar la gestión institucional. En particular, la Universidad Politécnica Salesiana ha desarrollado un plan estratégico para el período 2023-2025 centrado en la integración de tecnologías de la información en todos los procesos administrativos y académicos. Este plan trata sobre la automatización de procesos administrativos, poder utilizar herramientas de análisis de datos para la toma de decisiones, implementar plataformas virtuales para educación a distancia y poder desarrollar herramientas de gestión de espacios físicos.

Tenemos un caso muy conocido aquel caso es el desarrollado en la Universidad Politécnica de Madrid (UPM), donde López Velasco (López Velasco, 2024) documenta la creación de un sistema de reservas de aulas para actividades programadas. En su investigación, se identificó que la gestión manual de estos espacios podría tener muchos inconvenientes, entre ellos: asignaciones duplicadas, conflictos de horario y un uso ineficiente de los recursos disponibles. Este trabajo demuestra que la automatización de estos procesos no solo es posible, sino necesaria para optimizar la gestión de la infraestructura académica.

2.1.2 Transformación digital.

La reforma digital en instituciones de educación superior, según (Ávila Quiñónez, 2025) necesita añadir tecnologías digitales en todos los procedimientos de gestión académica y administrativa, así como en las formas de enseñanza, con el objetivo de mejorar la eficiencia operacional y la calidad educativa. De acuerdo con la Universidad Politécnica Salesiana (UPS, 2024), "la transformación digital no es únicamente de informatización, también es un cambio estructural que impacta en la cultura organizacional, la experiencia de usuarios y los modelos de gobernanza". En este escenario, implementar un sistema web para poder gestionar estos espacios universitarios es un paso definido dentro del proceso de transformación institucional.

John W. Branch nos enfatiza que la transformación digital en el ámbito de las instituciones de educación superior no son ajenas a estos retos de la transformación digital, que conlleva no solo a la adopción y adaptación de la misma sino también a un cambio cultural, generando sentido y valor para si mismas. Su mejora a través de la tecnología, no solo busca beneficiar a la institución sino también a la comunidad universitaria y la sociedad, ya que estos son el centro de existencia y escancia de una universidad (Branch, 2019)

2.1.3 Gestión de recursos físicos.

La gestión de recursos físicos en universidades abarca la planificación, uso, mantenimiento y control de espacios como aulas, auditorios, laboratorios y áreas deportivas. Un sistema eficiente de gestión debe permitir la trazabilidad del uso, evitar conflictos de disponibilidad y facilitar la programación estratégica de actividades dice (Futuver, 2025) Esto es bastante importante para lugares con alta demanda de espacios, como la Universidad Laica Eloy Alfaro de Manabí, donde la falta de coordinación genera ineficiencias al momento de que los usuarios deseen ocupar estos espacios.

2.1.4 Sistemas de información.

Los sistemas de información (SI) son conjuntos organizados de recursos tecnológicos y humanos que permiten recopilar, procesar, almacenar y distribuir información para apoyar la toma de decisiones y el control organizacional. En palabras de (Sandoval, 2016) "los SI son herramientas fundamentales para la mejora de la competitividad y la eficiencia organizacional en el contexto actual". El sistema propuesto en esta tesis corresponde a un SI

orientado a la gestión de escenarios, con funcionalidades de registro, control, notificación y trazabilidad.

2.1.5 Aplicaciones web.

Según (Pressman R. S., 2010) las aplicaciones web son sistemas de software que se ejecutan en navegadores web mediante una arquitectura cliente-servidor, permitiendo la interacción remota de los usuarios con los servicios ofrecidos por un servidor central. Este tipo de aplicaciones se diferencia de los sitios web tradicionales porque no solo presentan información, sino que ofrecen funcionalidades dinámicas e interactivas, similares a las de un programa de escritorio.

Según (Sommerville, 2011) una aplicación web es una colección de documentos y programas que residen en un servidor remoto y se acceden a través del internet utilizando un navegador. Estos sistemas suelen desarrollarse utilizando lenguajes de programación del lado del cliente como HTML, CSS y JavaScript y del lado del servidor como PHP, Python, Java o Node.js, y se comunican mediante protocolos como HTTP o HTTPS.

Por su parte, (Sommerville, 2011) clasifica las aplicaciones web dentro de los sistemas distribuidos, debido a que el procesamiento se divide entre el cliente (navegador) y el servidor. Este modelo nos ayuda mucho a que los usuarios interactúen con el sistema desde cualquier ubicación geográfica, claro siempre que este usuario pueda disponer de una conexión a Internet, lo que las convierte en una solución ideal para instituciones con comunidades extensas o distribuidas, como las universidades.

Destacan, (Doctor A. C., 2022) que el desarrollo de un sistema académico de información integrado basado en la web puede ayudar a los docentes de la Universidad Adamson a ser más efectivos y eficientes al aplicar exámenes, registrar las calificaciones, evitar redundancias de datos y esfuerzos, en sí que el tener un sistema web que pueda usarse en el contexto de la universidad, puede traer muchos aspectos positivos tanto como para los estudiantes, profesores y el personal administrativo.

En el contexto de la implementación de este proyecto, el desarrollo de una aplicación web para poder gestionar aquellos espacios físicos universitarios responde a la necesidad de superar los procesos manuales, fragmentados y poco trazables que actualmente limitan el uso eficiente de la infraestructura institucional. Esta solución permitirá a los usuarios tanto internos como externos a la comunidad universitaria acceder en tiempo real a funcionalidades como: consultas de disponibilidad, registro de solicitudes, seguimiento de aprobaciones y

reportes administrativos. Además, el carácter web de la aplicación asegura su compatibilidad multiplataforma (computadoras, tabletas y teléfonos), sin necesidad de instalar software adicional, facilitando su adopción y mantenimiento.

En términos técnicos, el proyecto adoptará una arquitectura cliente-servidor basada en tecnologías modernas como React para el frontend, Laravel para el backend y una base de datos relacional en PostgreSQL para persistencia. Este stack tecnológico se ha consolidado como una de las combinaciones más utilizadas para el desarrollo web moderno por su robustez, rendimiento y gran comunidad de soporte (Medeiros, 2022)

2.1.6 Tecnologías y arquitectura a emplear.

Las aplicaciones web como ya se ha mencionado anteriormente cuentan con una gran cantidad de herramientas y de amplio catálogo de tecnologías para la construcción de las mismas, para este proyecto se hará uso de tecnologías que están en tendencia, para asegurar una vida útil de la aplicación lo más extensa posible, como se divide en frontend y backend, una aplicación web, se optaron por las siguientes opciones.

Tabla 1. Tecnologías usadas en el Frontend.

Tabla categorizadora de tecnologías y enfoques para el Frontend.

Categoría	Tecnología	Justificación
Framework	ReactJS	Es un framework moderno para trabajar en el frontend de aplicaciones web.
Lenguaje de programación	JavaScript	Naturalmente es el que se usa en navegadores web.

Librerías complementarias	Axios, Zustand, TailwindCSS	Consumo de APIs, manejo de estados, estilización rápida.
Arquitectura adoptada	Clean Architecture (adaptada al cliente)	Mejora la mantenibilidad, separación de responsabilidades y escalabilidad del frontend.

Nota: En esta tabla se explica y detalla cada una de las tecnologías optadas para el desarrollo del frontend.

Tabla 2. Tecnologías usadas en el Backend.

Tabla categorizadora de tecnologías y enfoques para el Backend.

Categoría	Tecnología	Justificación
Framework	Laravel	Framework de PHP que nos ayuda en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación.
Lenguaje	PHP	Es un lenguaje de programación de código abierto que se usa normalmente en el desarrollo web, en el backend.
Arquitectura adoptada	Backend desacoplado	Nos ayuda en construir tanto frontend como backend de manera independiente y también funcionar así.
ORM	Eloquent ORM	ORM integrado de Laravel que permite trabajar con bases de datos usando modelos y relaciones de manera intuitiva, soporta migraciones y consultas seguras.
Base de datos	PostgreSQL	Es una base de datos relacional que tiene muy buena reputación además de un soporte de estándares técnicos.

Nota: En esta tabla se explica y detalla cada una de las tecnologías optadas para el desarrollo del backend.

2.2. Categorías Conceptuales

2.2.1 ReactJS.

Desarrollada por Facebook, ReactJS es una librería muy popular para el desarrollo de aplicaciones web, utilizada para construir interfaces de usuarios con alta interactividad y reactividad, especialmente para aplicaciones web de una sola página, single page application (SPA). Según (Medeiros, 2022) el modelo de ReactJS es basado en componentes, y su uso de Virtual DOM permite una actualización eficiente de la interfaz, mejorando significativamente la experiencia del usuario.

2.2.2 Axios.

Axios es una popular librería en el uso de aplicaciones web, es de código abierto y sirve para realizar peticiones HTTP entre aplicaciones, desarrollado sobre la base nativa de XMLHttpRequest nativo, este ofrece una solución más fácil y rápida de uso (geeksforgeeks, 2024).

2.2.3 Zustand.

Zustand librería pequeña, pero potente, para gestionar el estado en ReactJS. Fue creada por el equipo de Poimandres, zustand es conocido por su sencillez y eficacia, así como por eliminar la necesidad de boilerplate, lo cual nos da una opción moderna y eficaz comparada con soluciones más complejas como Redux(Poimandres, 2022).

2.2.4 TailwindCSS.

Bajo las palabras de Knowmad modo, Tailwindcss se describe como una potente herramienta para el desarrollo rápido en aplicaciones web, del lado del cliente, basándose en clases de utilidad que se aplican con facilidad dentro del código de HTML, permitiendo optimizar el peso del uso de CSS (knowmad mood, 2016).

2.2.5 Laravel.

Laravel es un framework de código abierto para desarrollo web, escrito en PHP y diseñado para seguir el patrón arquitectónico MVC (Modelo-Vista-Controlador). Ofrece un ecosistema completo con herramientas integradas como autenticación, migraciones, colas, testing y un sistema ORM elegante llamado Eloquent. Fue creado por Taylor Otwell en 2011

con el objetivo de simplificar el desarrollo de aplicaciones web robustas y eficientes. (Laravel, 2025).

2.2.6 PHP.

PHP es un lenguaje de programación de código abierto de scripts general que se enfoca particularmente al desarrollo web del lado del servidor. Es un lenguaje muy versátil y de múltiples plataformas, muy utilizado, que se emplea en Laravel para establecer la lógica backend de las aplicaciones. (arsys, 2025).

2.2.7 Eloquent ORM.

El sistema de mapeo objeto-relacional incorporado en Laravel es Eloquent ORM, que está basado en el patrón Active Record. Eloquent transforma filas y tablas de base de datos en clases PHP, o los llamados modelos, en situaciones de estas clases, posibilitando que se hagan consultas utilizando una sintaxis orientada a definir conexiones, objetos y soportar carga anticipada para poder prevenir dificultades de sentimientos, como el problema de N+1. (Cortés, 2024).

2.2.8 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto, reconocido por su estabilidad, potencia y conformidad con los estándares SQL. Es ampliamente utilizado en sistemas empresariales y académicos que requieren integridad transaccional, seguridad y escalabilidad (Stonebraker, The design of Postgres, 1990).

2.2.9 Clean Architecture.

El patrón de diseño de software propuesto por Robert C. Martin es Clean Architecture, que es un modelo que busca separar las responsabilidades de una aplicación en capas claramente definidas, tales como interfaces, casos de uso, entidades y frameworks. Esta perspectiva posibilita una gigantesca facilidad para ejecutar pruebas, independencia de tecnologías y sostenibilidad. (Martin, 2017).

2.2.10 Arquitectura Monolítica.

La arquitectura monolítica es un enfoque tradicional en el cual todos los componentes de una aplicación frontend, backend, lógica de negocio, base de datos están integrados y desplegados como una única unidad. Aunque limita la escalabilidad modular, ofrece

simplicidad en el despliegue y es adecuado para sistemas medianos con necesidades controladas (Richards, 2015).

2.3. Trazabilidad.

Según (Wohlrab, Pelliccione, Shahrokni, & Knauss, 2020) la trazabilidad es la habilidad para poder seguir el rastro de un requisito o componente desde su nacimiento hasta su utilización o implementación real final. Investigaciones recientes indican que la trazabilidad es imprescindible para manejar el cambio en la evolución de proyectos y el mantenimiento de nuestro software, ya que así podemos correlacionar requisitos, pruebas y código.

La trazabilidad asegura que tengamos transparencia a nivel institucional al ofrecer una visibilidad precisa del flujo de información, quién hizo qué, cuándo y por qué, lo cual nos facilita la auditoría y consolida los procedimientos de control interno. (Wohlrab, Pelliccione, Shahrokni, & Knauss, 2020).

En contextos complejos, como los de la industria 4.0, se emplean registros distribuidos y blockchain para robustecer la trazabilidad interorganizacional, lo cual nos sirve como referencia metodológica para sistemas universitarios con varios participantes.

2.4. Sistema de reserva

Mediante un sistema web de reservas podemos elegir recursos físicos como auditorios, laboratorios o aulas. Nuestro sistema verificaría la disponibilidad, manejaría las disputas temporales y aceptaría o negaría las solicitudes mediante los administradores.

Un caso auténtico de una universidad en Indonesia, el cual sucedió en 2022 evidenció que después de incorporar reservas automatizadas para los salones de estudio y laboratorios, se pudieron conseguir avances del 30% en la eficiencia. Los sistemas digitales, en comparación a los métodos manuales, disminuyen el tiempo de gestión en más del 40%, previenen la duplicación de reservas y posibilitan el acceso las 24 horas del día, los siete días de la semana, con incluso rastreo de uso. Tener integraciones de IoT en el sistema de reserva, sería un plus que posibilitaría la vigilancia de las condiciones ambientales y la ocupación. (Abrahamsson, 2003).

2.5. Marco Legal

2.5.1 Introducción

El presente capítulo tiene como objetivo fundamentar jurídicamente el estudio sobre la optimización de la gestión, control y trazabilidad de los escenarios universitarios en la

Universidad Laica Eloy Alfaro de Manabí (ULEAM). Se busca establecer el respaldo normativo institucional y legal que garantiza la importancia de una administración eficiente de los recursos físicos y la necesidad de modernizar los procesos a través de la transformación digital, un eje fundamental para cualquier institución educativa. La implementación de una aplicación web para la gestión de estos espacios se alinea con las exigencias tecnológicas actuales y los principios de buena gobernanza en el ámbito educativo.

2.5.2 Bases Constitucionales

La Constitución de la República del Ecuador establece los principios esenciales que sostienen la importancia de una administración pública que sea eficiente y transparente, incluidas las instituciones universitarias. A pesar de que no se refieren directamente a la administración de escenarios, los artículos que se presentan a continuación son significativos para enmarcar legalmente la investigación.

- **Artículo 26:** Establece que “la educación es un derecho de los individuos durante toda su vida y una obligación irrenunciable e inexcusable del Estado”. Esto pone de relieve la relevancia de una gestión universitaria que asegure las condiciones ideales para el ejercicio de este derecho, con lo cual se incluye el acceso y la utilización correcta de la infraestructura.
- **Artículo 344:** Dictamina que el sistema de educación superior tiene la obligación de asegurar “la calidad, la pertinencia, la igualdad de oportunidades y la gratuidad hasta el tercer nivel”. Una administración eficaz de las situaciones ayuda a garantizar la calidad y la igualdad en el acceso a los recursos académicos y recreativos, lo que fomenta un entorno educativo apropiado.

2.5.3 Legislación Nacional.

Las leyes nacionales, al desarrollar los principios de la constitución, ofrecen un marco más concreto para el manejo público y la educación superior:

- **Ley Orgánica de Educación Superior (LOES):** La organización y el funcionamiento del sistema de educación superior en Ecuador están regidos por esta ley. La LOES, en general fomenta la gestión responsable de los recursos, la calidad y la eficiencia en las Instituciones de Educación Superior (IES), a pesar de que el ejemplo dado se enfoca en becas. Esta legislación, que tiene como objetivo perfeccionar los procedimientos operativos y administrativos en las universidades, respalda la exigencia de

modernización y empleo de tecnologías informáticas en la administración institucional, tal como lo determina la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación. (Senecyt, 2022)

2.6. Conclusiones del marco teórico.

Durante el desarrollo del marco teórico, se han tratado los conceptos clave que respaldan la factibilidad y necesidad del proyecto propuesto. La transformación digital, que es un proceso completo de modernización en las instituciones, ha sido resaltada como un elemento crucial en el ámbito actual de las instituciones educativas, particularmente en una situación que requiere eficacia, trazabilidad y optimización de recursos. En esta línea se demostró que implementar sistemas informáticos enfocados en la administración de recursos ficticios, como los espacios universitarios, es una medida crucial para sobrepasar las restricciones originadas por procedimientos manuales y fragmentados.

Las aplicaciones web y en general los sistemas de información, se han establecido como opciones tecnológicas adecuadas y eficaces para simplificar la comunicación entre usuarios, automatizar los procedimientos de reserva y centralizar la información operativa de las entidades. Estos sistemas no solo posibilitan una mejor supervisión del uso y disponibilidad de los espacios, sino que además refuerzan la transparencia y las decisiones basada en datos, contribuyendo de esta manera a una administración más moderna y fundamentada en buenas prácticas.

Desde un enfoque técnico, hemos fundamentado el porqué de la selección del stack que vamos a trabajar, compuesto por React, Laravel y PostgreSQL, que al mismo tiempo la complementa un tipo de estructura basada en los principios de Clean Architecture. Tener esto nos asegurará que el sistema sea mantenible, escalable y con un repartimiento de responsabilidades bien definido, lo cuál nos ayudará a poder escalarlo en el futuro. De igual manera hemos reconocido la importancia de utilizar metodologías ágiles en situaciones en las que los requisitos del cliente puedan ser diferentes durante todo el proceso de desarrollo.

En general, las teorías analizadas no solo fortalecen el planteamiento del problema, sino que también guían los métodos propuestos para el proyecto. Esto demuestra que crear un sistema web para la administración y gestión de contextos universitarios es una solución válida, esencial y respaldada a nivel académico, tecnológico e institucional.

Capítulo III: Metodología

3.1. Introducción

El enfoque, los métodos y las técnicas que se utilizarán para la recopilación y análisis de datos requeridos en el desarrollo e implementación de una aplicación web destinada a gestionar espacios universitarios en la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta son descritos en este capítulo, La metodología adoptada se ajusta tanto al problema detectado como a la fundamentación teórica desarrollada previamente, lo que asegura un cimiento firme para lograr las metas del proyecto.

Este tipo de estudio se puede definir como la aplicación de una investigación que contiene un enfoque descriptivo y tecnológico. Este estudio es aplicado ya que se está buscando resolver una problemática sobre la gestión de los escenarios usando la tecnología, más en concreto mediante una aplicación web, y además decimos que es descriptiva porque estamos analizando el estado actual de los procesos en la universidad para poder identificar aquellos requisitos funcionales del sistema propuesto.

3.2. Catálogo de partes interesadas y actores

Para tener bien definido el alcance que tendrá el sistema y asegurar de que este mismo cumpla con las necesidades que tiene la universidad, identificamos a una población objetivo que incluye a las partes interesadas en este proceso de gestión de escenarios.

La población y la muestra están compuestas de:

- **Personal Administrativo:** Administradores encargados de la gestión, aprobación y control de los escenarios. Se tiene previsto una muestra de 3 administradores.
- **Docentes:** Usuarios internos de la universidad que sean docentes y requieran espacios para actividades académicas. Se tiene previsto una muestra de 10 docentes.
- **Estudiantes:** Estudiantes de la ULEAM que utilicen frecuentemente los espacios que hay por toda la universidad.

- **Usuarios Externos:** Individuos o entidades que no son parte de la comunidad universitaria y necesitan usar un espacio a través de una solicitud formal. Se tiene previsto una muestra de 5 usuarios externos.

Respecto a estos usuarios externos identificados, cabe aclarar que la fórmula de muestreo probabilístico se aplicó exclusivamente a la población estudiantil, debido a que representan la demanda principal.

3.3. Recolección de Datos.

Para poder realizar el levantamiento de información y de requisitos, vamos a implementar una estrategia de análisis mixta (cualitativa y cuantitativa), para poder lograr esto vamos a utilizar los siguientes métodos, técnicas e instrumentos.

3.4. Población y Muestra.

La validación del sistema se hizo con la consideración de una población de 500 estudiantes de la matriz Manta. El tamaño de muestra que se escogió de nuestra población se determinó mediante la fórmula para poblaciones finitas de (Aguilar-Barojas, 2005), aplicando un nivel de confianza del 95% y un margen de error del 13.1% para ajustarse a los recursos disponibles para la fase de pruebas.

3.4.1 Fórmula aplicada.

$$n = \frac{Z^2 \cdot p \cdot q \cdot N}{e^2 \cdot (N - 1) + Z^2 \cdot p \cdot q}$$

Donde:

- **N (población):** 500 estudiantes.
- **Z (Nivel de confianza):** 1.96 (95%).
- **e (Error de muestra):** 0.131 (13.1%).
- **p y q (Probabilidad):** 0.5.

El cálculo de la fórmula de muestra nos dio que la muestra representativa es de 50 estudiantes, quienes participarán en la recolección de datos mediante las encuestas estructuradas.

3.5. Métodos de investigación.

3.5.1 Métodos Teóricos.

- **Análisis Documental:** Procedimos con la revisión de la bibliografía y normativa institucional que está relacionada con el cambio tecnológico, desarrollo de sistemas y gestión de recursos.
- **Análisis de conceptos:** Fundamentación del progreso del sistema web a través de elaborar el marco teórico.

3.5.2 Métodos Empíricos.

- **Observación en Directo:** Recopilamos datos acerca de cómo es el proceso de reservas de los espacios de la ULEAM.
- **Reuniones Guías:** Detección de necesidades funcionales y no funcionales a través de reuniones guías que tuvimos con nuestro tutor y personal administrativo.
- **Encuestas:** Se la aplicamos a los alumnos, docente, personal administrativo de la universidad y usuarios externos.

3.6. Técnicas e Instrumentos

Para poder ejecutar los métodos empíricos, se utilizarán las siguientes técnicas:

3.6.1 Técnica: Encuesta estructurada.

- **Instrumento:** Se llevaron a cabo encuestas con preguntas cerradas de tipo Likert (especificado en el **Anexo 1**), se utilizó a través de Google Forms para alumnos, docentes, personal administrativo y usuarios externo. La meta es ser capaz de recopilar la opinión actual acerca del sistema y la disposición hacia una solución digital.

3.6.2 Técnica: Entrevista semiestructurada.

- **Instrumento:** Guía por parte del tutor y personal administrativo (especificada en el **Anexo 2**) de la universidad, en el cuál tuvimos reuniones sobre el desarrollo del sistema web, dónde nos brindaron sus necesidades hacía lo que tiene que tener el producto.

3.7. Levantamiento de Procesos del Negocio (As-Is)

Una de las fases críticas es el levantamiento de los procesos actuales del negocio, qué en este caso sería gestión de escenarios, se alimenta directamente de los métodos empíricos descritos anteriormente.

El objetivo es poder comprender el flujo de trabajo existente para poder modelar una solución (To-Be) que pueda optimizarlo. Para poder lograrlo lo hacemos mediante:

1. **Observación Directa y No Participante:** Utilizamos esta para poder documentar los flujos reales del proceso de reserva. Mediante la ficha de observación podemos registrar los actores, las acciones que realizan y los documentos que pueden llegar a utilizar:
2. **Entrevistas Semiestructuradas:** Mediante estas el personal administrativo y docente pueden describir el proceso actual desde su perspectiva, poder identificar aquellas dificultades, y poder validar los nuevos requerimientos del sistema.

3.8. Descripción de la Metodología de Desarrollo y Selección de Artefactos

Para la construcción de la aplicación web, se ha utilizado una metodología de desarrollo ágil llamada Programación Extrema (XP).

3.8.1 Justificación de la Metodología (XP)

La elección de XP se justifica debido a su alineación con los objetivos del proyecto y el contexto de la tesis:

- **Enfoque en Calidad:** Al usar XP se promueve prácticas que garantizan un código de alta calidad, usando las pruebas unitarias y la refactorización. Esto es esencial para poder asegurar la mantenibilidad del backend en Laravel y el frontend en React.
- **Adaptabilidad y Retroalimentación:** Al usar la metodología que elegimos nos enfocamos en ciclos de desarrollo breves que ayudan a que tengamos entregas funcionales y constantes. Teniendo la validación anticipada por parte de expertos y tutores, lo que nos ayuda a poder implementar notificaciones sin demoras.

- **Simplicidad y Diseño Evolutivo:** Al usar XP podemos mantener el diseño simple, permitiendo que la arquitectura evolucione de manera natural, esto es muy ideal para un proyecto que parte de objetivos específicos pero que se beneficia bastante de mejoras incrementales.
- **Comunicación:** Los principios de comunicación de XP son necesarias para poder interaccionar entre los líderes de los proyectos.

3.8.2 Selección de Artefactos y Prácticas de XP

Aunque utilizar la metodología ágil nos da camino a un conjunto amplio de prácticas, en el contexto de este proyecto se priorizarán los siguientes artefactos y prácticas clave:

- **Historias de Usuario:** Aquellas necesidades funcionales que se recolectaron de las entrevistas y encuestas se traducirán en historias de usuario para planificar las iteraciones.
- **Ciclos Cortos (Iteraciones):** Debido a que es XP este proyecto se dividirá en entregas pequeñas y funcionales como se detalla en el cronograma (**Tabla 4**).
- **Pruebas Unitarias (TDD):** Utilizaremos el Desarrollo Orientado a Pruebas (TDD), asegurando de que cada funcionalidad, tanto en el backend como en el frontend tenga pruebas unitarias.
- **Refactorización Continua:** Podremos mantener la calidad del código mediante la refactorización constante, asegurando el cumplimiento de estándares de codificación, PSR-12 para Laravel y guías de estilos para React.
- **Diseño Simple y Evolutivo:** Comenzamos con el diseño más simple y permitiendo que toda la arquitectura que está basada en Clean Code evolucione para poder evitar la sobreingeniería.

Capítulo IV: Propuesta o Desarrollo del Proyecto

4.1. Introducción

El capítulo a continuación presenta la propuesta técnica del sistema describiendo la metodología de desarrollo, el diseño de base de datos, la arquitectura y las primeras implementaciones funcionales, siendo un poco más detallado en la propuesta y el desarrollo integral del “Sistema Web para la Gestión de Escenarios de la Universidad Laica Eloy Alfaro de Manabí (ULEAM)”. En esta fase vamos a poder materializar los fundamentos teóricos y metodológicos expuestos en los capítulos anteriores, teniendo como evidencia el tránsito crítico desde el diseño conceptual abstracto hasta la implementación funcional y tangible de esta solución tecnológica robusta.

El cometido fundamental de esta propuesta técnica supera la simple automatización de tareas; se quiere demostrar la viabilidad, robustez y funcionalidad de un aplicativo web para poder mejorar en la gestión de los escenarios, el control administrativo y la trazabilidad de los escenarios universitarios. Esta solución puede abordar estructuralmente la problemática de la descentralización, la gestión manual que tiende a tener errores y así mismo la pérdida de la información histórica, mediante una arquitectura moderna, modular y escalable. Este sistema integra un frontend altamente interactivo, centrado en la experiencia de usuario, con un backend que así mismo es seguro y eficiente, garantizando de esa manera la integridad de los datos transaccionales, tener seguridad en el control de acceso y una operatividad fluida que nos permite tener menos carga operativa del personal administrativo.

La justificación del enfoque tecnológico adoptado, que está fundamentado en el stack React-Laravel-PostgreSQL, radica que tenemos la necesidad imperiosa de construir un ecosistema digital resiliente. Este de aquí no solo debe responder a los requerimientos funcionales inmediatos de la ULEAM, sino también poder ofrecer garantías de escalabilidad horizontal y mantenibilidad a largo plazo. Como señalan (Pressman R. S., Software Engineering: A Practitioner's Approach, 2020), la elección de una arquitectura basado en componentes reutilizables y servicios desacoplados es crucial para el desarrollo de software evolutivo en entornos institucionales complejos, donde las reglas de negocio son bastantes susceptibles a cambios tanto normativos o administrativos.

De igual manera, la adopción de la metodología de Programación Extrema (XP) permita tener una adaptación ágil a la incertidumbre inherente a los procesos de investigación

y desarrollo, de esta manera podemos tener de manera asegurada una entrega continua de valor alineada estratégicamente con las políticas de modernización digital promovidas por la (Senecyt, 2022)

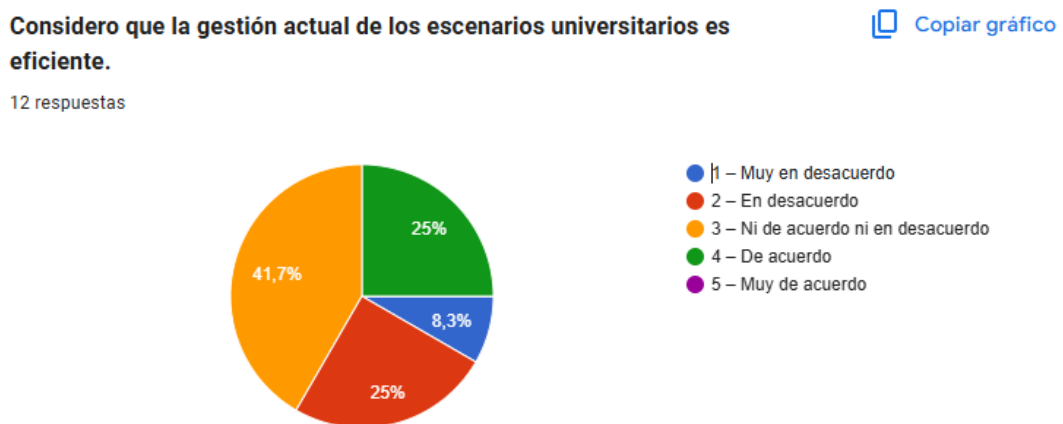
4.2. Análisis del Diagnóstico

Para poder fundamentar el desarrollo técnico de este sistema, se realizó un levantamiento de información mediante encuestas digitales. El objetivo era validar nuestra justificación al momento de querer realizar este proyecto mediante la percepción de los usuarios sobre la gestión actual y su apertura hacia una solución tecnológica. Si bien la muestra definida fue de 50 estudiantes, en esta fase se analizaron a 12 respuestas correspondientes a un estudio piloto, cuyo objetivo fue validar la viabilidad del sistema, como veremos a continuación

4.2.1 Evaluación de la gestión actual

Pregunta: Considero que la gestión actual de los escenarios universitarios es eficiente.

Figura 2. Cuál es la percepción de la gestión actual.



Nota: Autoría Propia.

Lo primero que realizamos fue analizar que aquellos procesos que son de manera manual aún vigentes cumplen con las expectativas de los usuarios. Los datos muestran una insatisfacción importante: solo el 25% de los encuestados considera que la gestión de los escenarios es eficiente, mientras que un 33,3% manifiesta un desacuerdo directo.

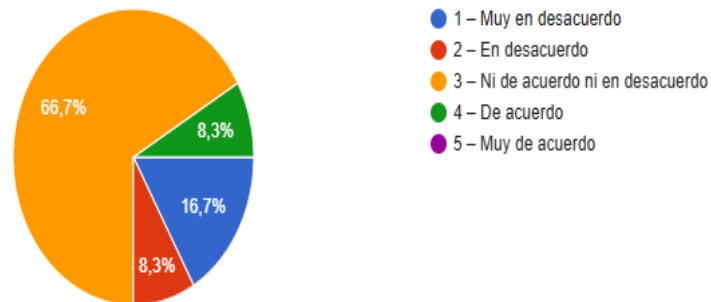
Pregunta: El proceso actual para solicitar un espacio físico en la universidad es fácil de entender.

Figura 3. Es fácil entender el proceso actual.

El proceso actual para solicitar un espacio físico en la universidad es fácil de entender.

[Copiar gráfico](#)

12 respuestas



Nota: Autoría Propia.

Aquí en esta pregunta los resultados muestran una confusión considerable, ya que apenas el 8.3% de los encuestados están de acuerdo con la facilidad del proceso. Tener en cuenta del que el 25% está en desacuerdo y un 66.7% se mantenga neutral refleja que el método tradicional carece de claridad necesaria para el usuario común.

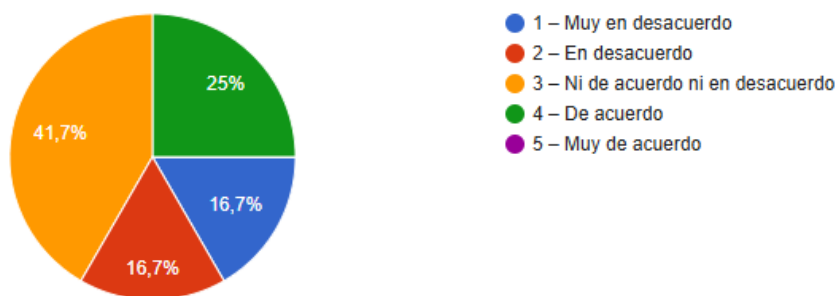
Pregunta: La información sobre los pasos para reservar una aula o auditorio es clara y accesible.

Figura 4. Gráfico sobre si los pasos para reservar un espacio son claros.

La información sobre los pasos para reservar un aula o auditorio es clara y accesible.

 Copiar gráfico

12 respuestas



Nota: Autoría Propia.

Observamos que tenemos una fragmentación en la información, donde un 33.4% de los usuarios nos dice que los pasos no son claros. Esto nos da a entender que los usuarios tienen una dependencia de la atención presencial para resolver dudas, ralentizando la gestión administrativa.

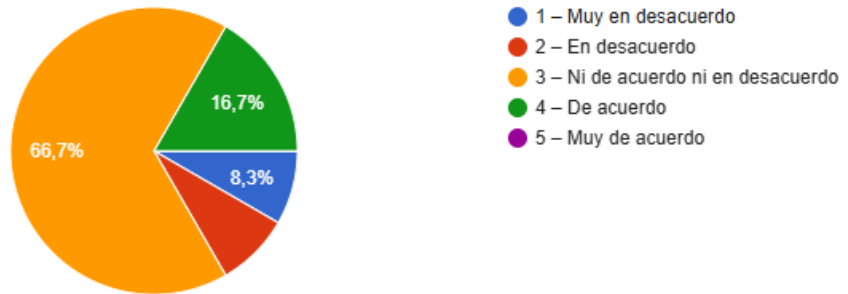
Pregunta: He tenido problemas o demoras al intentar reservar un escenario universitario.

Figura 5. Gráfico sobre si hay problemas o demoras al querer reservar un espacio.

He tenido problemas o demoras al intentar reservar un escenario universitario.

 Copiar gráfico

12 respuestas

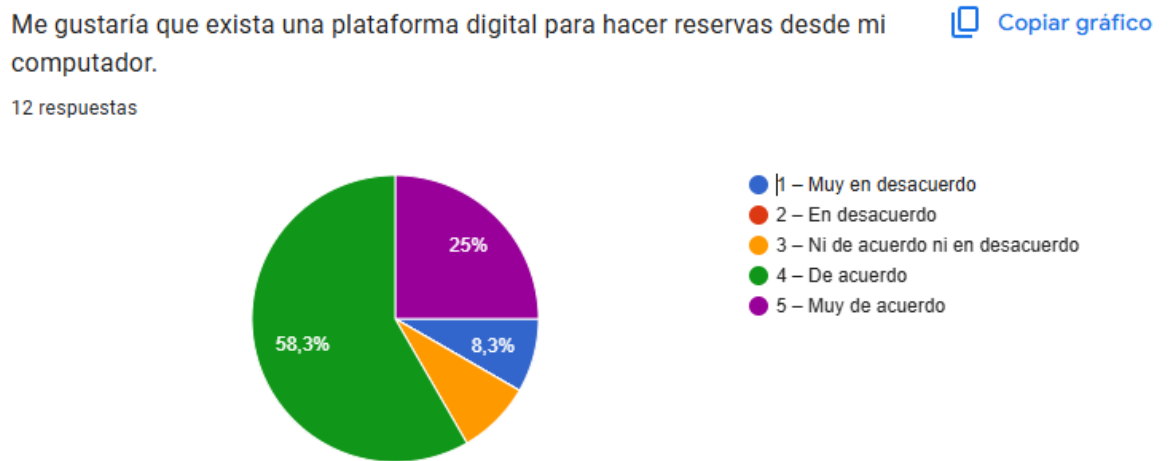


Nota: Autoría Propia.

Un 16.7% de los encuestados confirma haber enfrentado demoras activas. Aunque el porcentaje de neutralidad es alto 66.7%, la existencia de problemas reales valida la necesidad de poder implementar un sistema para poder resolver estos problemas y elimine los tiempos muertos.

Pregunta: Me gustaría que exista una plataforma digital para hacer reservas desde mi computador.

Figura 6. Preferencia sobre existencia de una plataforma digital.



Nota: Autoría Propia.

Los resultados que podemos ver nos indican que hay una gran aceptación hacia la creación del sistema dentro del grupo encuestado, un 83% de los usuarios que se encuestaron si les gustaría una plataforma digital para hacer reservas desde un computador.

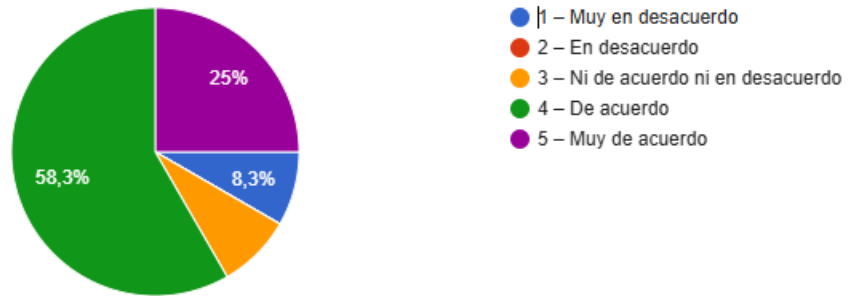
Pregunta: Un sistema web facilitaría el proceso de solicitud y aprobación de escenarios universitarios.

Figura 7. Requerimientos de un sistema web.

Un sistema web facilitaría el proceso de solicitud y aprobación de escenarios universitarios.

 Copiar gráfico

12 respuestas



Nota: *Autoría Propia.*

Tener una agilización con un sistema web es algo que se desea teniendo en cuenta el 83.3% de favorabilidad. Los usuarios saben que un sistema web no solo simplifica la solicitud, sino que facilita la tarea de aprobación para los administradores.

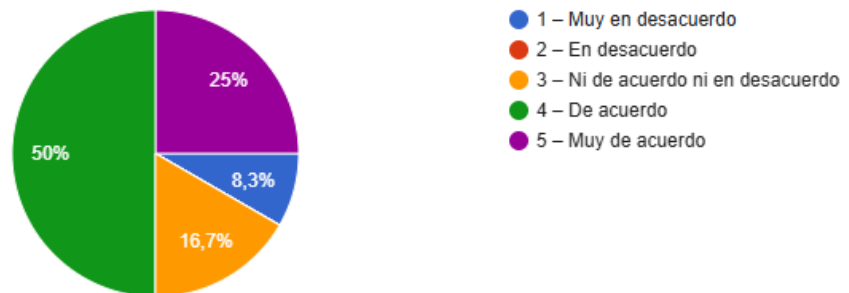
Pregunta: Me sentiría más seguro/a utilizando un sistema digital con historial y seguimiento de solicitudes.

Figura 8. Los usuarios se sentirían seguros con historial y seguimiento.

Me sentiría más seguro/a utilizando un sistema digital con historial y seguimiento de solicitudes.

 Copiar gráfico

12 respuestas



Nota: Autoría Propia.

La trazabilidad es algo que, si requieren los usuarios, teniendo como un 75% de usuarios que si desean tener una trazabilidad. Contar con un historial digital genera mucha más confianza que algo de manera física no tendría.

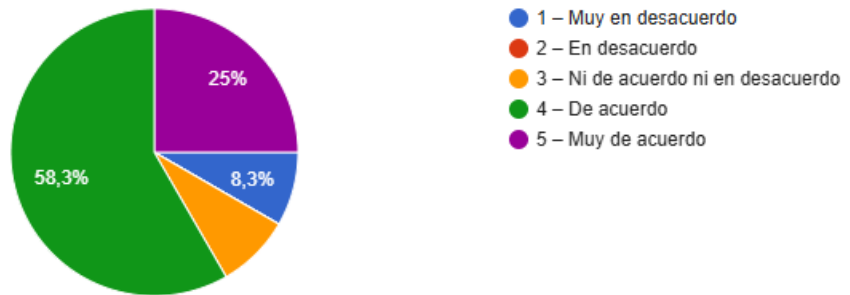
Pregunta: Considero que una aplicación web mejoraría la transparencia en la asignación de escenarios universitarios.

Figura 9. Mejoraría la transparencia en la asignación de espacios con un sistema web.

Considero que una aplicación web mejoraría la transparencia en la asignación de escenarios universitarios.

 Copiar gráfico

12 respuestas



Nota: Autoría Propia.

Existe una muy buena aceptación, un total de 83.3% de los usuarios sobre que una aplicación web mejoraría la transparencia en la asignación de escenarios. Esto valida el impacto institucional positivo que busca este proyecto de tesis.

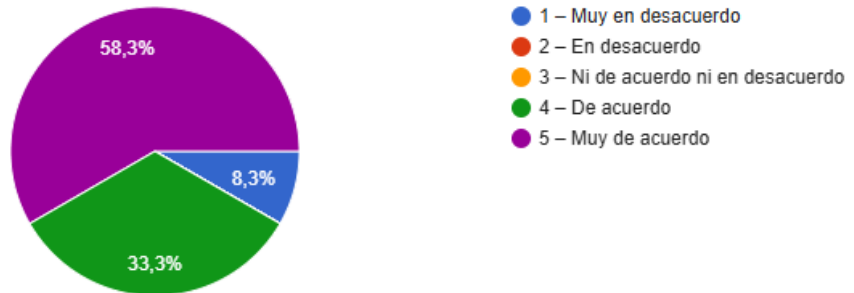
Pregunta: Recomendaría el uso de una plataforma web para reservar escenarios universitarios a otros usuarios.

Figura 10. Usuarios recomendarían la plataforma web.

Recomendaría el uso de una plataforma web para reservar escenarios universitarios a otros usuarios.

 [Copiar gráfico](#)

12 respuestas



Nota: *Autoría Propia.*

Los resultados dentro de esta pregunta evidencian que la mayoría de usuarios recomendarían el uso de una plataforma para reservar escenarios: un 91.6% para ser exactos, este resultado asegura que la adopción del sistema será rápida y orgánica para todos.

4.3. Metodología de desarrollo seleccionada

En el desarrollo del sistema se aplicó rigurosamente la metodología ágil, Programación Extrema (XP).

Al momento de elegir esta metodología no fue de manera arbitraria, sino que responde a la naturaleza dinámica y exploratoria de un proyecto de tesis, dónde los requerimientos que definimos al principio suelen evolucionar conforme se profundiza en el entendimiento de las necesidades institucionales. A diferencia de otras metodologías predictivas o en cascada, que normalmente necesitan de requerimientos inmutables desde el principio, XP abraza el entorno y el cambio como una ventaja competitiva. Según (Beck, 2004), XP tiene sus bases en cuatro valores fundamentales: simplicidad en el diseño, comunicación constante con el cliente, retroalimentación rápida a través de entregas frecuentes y la valentía para refactorizar código cuando es necesario.

Implementar la metodología XP en este proyecto se caracterizó debido a la adopción de prácticas de ingeniería de software disciplinadas, priorizando que tengamos bastante de calidad de código sobre la documentación exhaustiva no funcional. La organización de este trabajo se estructuró en iteraciones cortas, para ser más específicos en Sprints de 2 semanas y ciclos de desarrollo rápidos, lo que permitió realizar entregas parciales funcionales, al final de cada ciclo. Este enfoque iterativo e incremental facilitó la detección temprana de errores lógicos y de diseño, así de esa manera poder asegurar que el producto final estuviera perfectamente alineado con las expectativas de los interesados desde las etapas iniciales del desarrollo.

Figura 11. Iteraciones Dentro del Backend.



Nota: Implementación de las iteraciones cortas de acuerdo a la metodología elegida, Programación Extrema (XP).

4.4. Organización de Iteraciones y Roles

Debido a que el proyecto se desarrolla en el contexto académico de una tesis de grado o memoria escrita, la asignación de roles tradicionales de XP se adaptó para maximizar la eficiencia de los recursos disponibles, manteniendo la esencia de la metodología:

- **Desarrolladores FullStack (Castillo Vera Ariel & Cedeño Mosquera Jostin):** Asumieron la responsabilidad integral de la implementación técnica, abarcando desde el diseño de la base de datos y la lógica del servidor (backend) hasta la construcción del cliente y su interfaz de usuario (frontend).
- **Cliente (Tutores y Directivos):** Actuaron como la voz del negocio, la validación funcional fue realizada por docente tutores mediante revisión de flujos y pruebas de uso, definieron las prioridades que tenían así de esa manera llegamos a una conclusión de los requisitos, nos clarificaron las dudas que teníamos sobre los procesos institucionales y proporcionando retroalimentación crítica (User Acceptance Testing).

- **Tracker:** Utilizamos herramientas para poder gestionar las tareas para poder monitorear las métricas y el progreso del cronograma, garantizando así poder llegar a alcanzar hitos académicos dentro de la institución.

El ciclo de vida de desarrollo que hicimos siguió las características de XP identificadas por (Pressman R. S., 2010), ejecutadas de manera cíclica y repetitiva:

1. **Exploración (Análisis y Levantamiento):** Tenemos la definición detallada de historias de usuarios basados en los requisitos levantados en el Capítulo III. En aquella fase se tuvo en claro las reglas del negocio complejas, como la prohibición de solapamiento de horarios y la jerarquía de privilegio de roles.

Tabla 3. Historia de Usuarios y Criterios de aceptación.

Tabla dónde se ubican las historias de usuarios.

ID	Tipo	Historia de Usuario	Criterios de aceptación.
HU-01	Funcional.	Como admin quiero gestionar los espacios universitarios para mantener el inventario institucional actualizado.	El sistema debe mostrar un formulario para registrar atributos del espacio. Debe requerir campos obligatorios antes de guardar. Debe permitir modificar el estado.
HU-02	Funcional.	Como usuario interno o externo, quiero poder visualizar la disponibilidad de los escenarios.	Mostrar un calendario o listado interactivo. Bloquear visualmente las fechas y horas que ya cuenten con una reserva aprobada.

HU-03	Funcional.	Como usuario, quiero enviar una solicitud de reserva indicando fecha, hora y motivo para usar un escenario.	El sistema debe validar que el horario elegido no esté ocupado. La solicitud debe guardarse con el estado inicial de “Pendiente” hasta ser revisada.
HU-04	Funcional.	Como administrador, quiero aprobar o rechazar las solicitudes de reserva para llevar el control del uso de las instalaciones.	Visualizar una bandeja de solicitudes pendientes. Permitir cambiar el estado de “Aprobado” o “Rechazado”. Registrar quién realizó la acción.
HNF-01	No funcional.	Como sistema, requiero que el acceso a la información sea segura y restringida mediante roles.	Se debe implementar autenticación. Un usuario normal no va a poder acceder a rutas de administrador.

HNF-02	No funcional.	Como sistema, requiero que la interfaz sea adaptable, o responsive para permitir el acceso desde distintos dispositivos.	La interfaz web debe ajustarse correctamente a pantallas de celular, tablet y escritorio.
---------------	---------------	--	---

Nota: Tenemos las distintas historias de usuarios funcionales y no funcionales.

2. **Planificación (Planning Game):** Aquí estimamos el esfuerzo técnico para cada historia de usuario y selección estratégica de las funcionalidades a implementar en la siguiente iteración, poniendo de primer lugar aquellas que aportaban mayor valor inmediato o mejor dicho las MVP.
3. **Iteraciones (Diseño/Codificación/Pruebas):** Utilizando el Test Driven Development (TDD) desarrollamos los módulos, estudios empíricos demuestran que usar TDD ayuda a mejorar de manera muy exponencial la calidad del código y reducir la densidad de efectos en producción, aunque pueda requerir un mayor esfuerzo inicial del desarrollo. (Janzen, 2006)
4. **Puesta en producción (Entrega y Retroalimentación):** Puesta a prueba de los módulos funcionales en un entorno de pruebas (Staging) para la revisión y validación por parte del cliente.

4.5. Cronograma de Desarrollo Detallada

Ahora se va a presentar el cronograma ejecutado para la fase de resultados. Este plan de trabajo refleja un enfoque iterativo incremental, abarcando desde el análisis fundacional hasta el despliegue de la versión beta operativa:

Tabla 4. Cronograma de actividades.

Tabla de iteraciones y despliegues.

Etapas	Actividad	Tiempo de Duración (Semanas)	Lo que se entrega y especificaciones técnicas.
Análisis y diseño.	Requerimientos y modelado fundamental de datos.	Semanas 1-2	Modelo Entidad y Relación y Diagramas de Flujo: Definición de entidades, relaciones y reglas de integridad referencial.
Configuración	Setup de entornos de desarrollo (Laravel y React)	Semana 3	Repositorios y “Hola Mundo”: Configuración del entorno e inicialización de los 2 repositorios de GitHub.

Iteración 1	Módulo de Seguridad (Auth, Roles y Permisos)	Semana 4-5	API Auth y Gestión de Usuarios: Autenticación mediante Laravel Sanctum usando tokens.
Iteración 2	Inventario de Espacios y Mobiliario	Semanas 6-8	CRUD de Escenarios: Desarrollo de interfaces para gestión de aulas y auditorios con carga asíncrona de imágenes.
Iteración 3	Motor de Reservas y Solicitudes	Semanas 9-11	Core del Negocio: Flujo completo desde solicitud_espacio hasta la confirmación en booking, con algoritmos de validación.
Iteración 4	Servicios Adicionales y Reportes	Semanas 12-13	Valor Agregado: Desarrollo del módulo de Tarifas y Dashboard estadístico de ocupación.

Pruebas y Despliegue	Pruebas de integración y corrección de errores	Semanas 14-15	Versión Beta Funcional: Ejecución de baterías de pruebas y despliegue final en servidor de pruebas.
-----------------------------	--	---------------	---

Nota: En esta tabla se explica cada una de las iteraciones hasta la versión beta funcional.

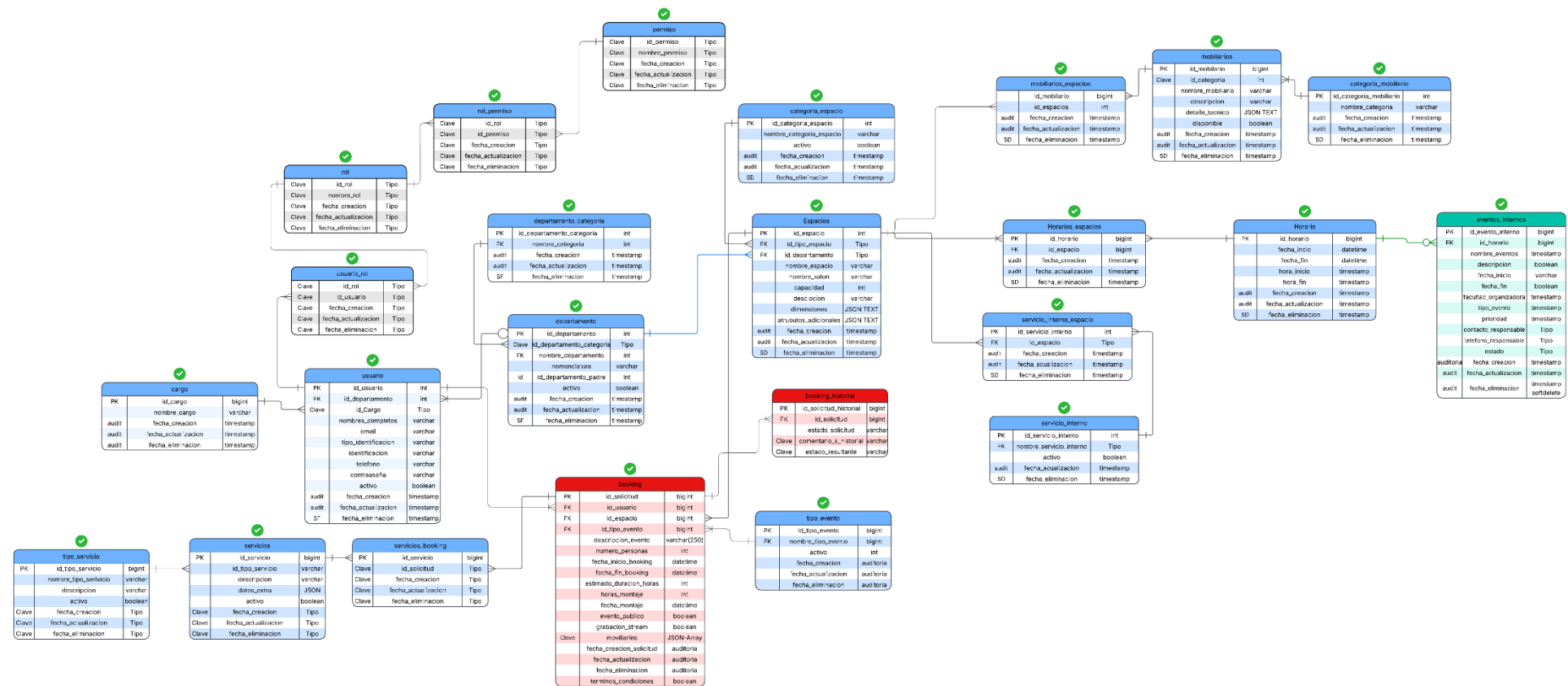
4.6. Diseño de la base de datos.

El diseñar la base datos constituye en la columna vertebral del sistema web, este siendo un componente indispensable y crítico para el desarrollo, garantizar la integridad referencial, la consistencia atómica de las transacciones y la eficiencia en las consultas complejas. La base de datos se diseñó utilizando criterios de tercera forma normal (3FN), separando entidades independientes con el objetivo de eliminar la redundancia de datos y prevenir anomalías de actualización, inserción o borrado (Pressman R. S., Ingeniería de Software: Un enfoque práctico (9.^a ed.). McGraw-Hill Education., 2020)

Se implementó la base de datos sobre el motor de base de datos PostgreSQL. Justificamos esta elección debido a su robustez comprobada en entornos empresariales y su diseño orientado a objetos que soporta tipos de datos complejos. Según (Stonebraker, The design of POSTGRES, 1986) el diseño de PostgreSQL permite extender el modelo relacional tradicional para soportar reglas complejas y tipo de datos abstractos, lo cual es ideal para manejar la lógica de reservas y disponibilidad. Además, su soporte nativo para JSONB permite una flexibilidad híbrida (NoSQL) dentro del modelo relacional para atributos cambiantes, como las especificaciones técnicas del mobiliario (Momjian, 2000)

A continuación, se presenta el Modelo de Entidad-Relación (E-R) del sistema, el cual refleja fielmente la estructura operativa y las reglas de negocio de la gestión de escenarios en la ULEAM:

Figura 12. Diagrama Entidad-Relación de la base de datos del sistema.



Nota: Esta figura representa al diseño de base de datos mediante el modelo de Entidad-Relación.

4.7. Descripción Detallada de las Entidades Principales y sus Relaciones

El diagrama consiste en cuatro módulos esenciales, que están diseñadas para poder funcionar de formula modular pero bien interconectada:

1. Módulo de Seguridad y Acceso (RBAC):

- **usuario:** Tabla main que contiene los datos a nivel personal e institucional.
- **rol y permiso:** Establecen un sólido sistema de Control de Acceso Basado en Roles. Este modelo, establecido por (R. S. Sandhu, 1996), facilita la gestión de seguridad al otorgar permisos de roles en vez de a usuarios individuales, lo que posibilita una administración escalonada de privilegios.
- **cargo y departamento:** Tablas que contienen parámetros para tener una clasificación institucional.

2. Módulo de Inventario de Espacios:

- **espacio:** Representa el espacio que se puede reservar, este de aquí guarda toda la información necesaria que tiene el espacio.
- **categoría_espacio:** Clasifica los espacios utilizando diferentes categorías para poder aplicar filtros.
- **mobiliario:** Administra los mobiliarios o equipamientos en cada espacio.

3. Módulo de Gestión de Reservas:

- **booking:** Entidad que va a estar documentando la confirmación de la solicitud del uso de un espacio.
- **solicitud_espacio:** Se encarga de hacer la “pre-reserva” del espacio para todos los usuarios.
- **evento_interno:** Posibilita la restricción de espacios para actividades propias de la institución.

4. Módulo de Servicios y Tarifación:

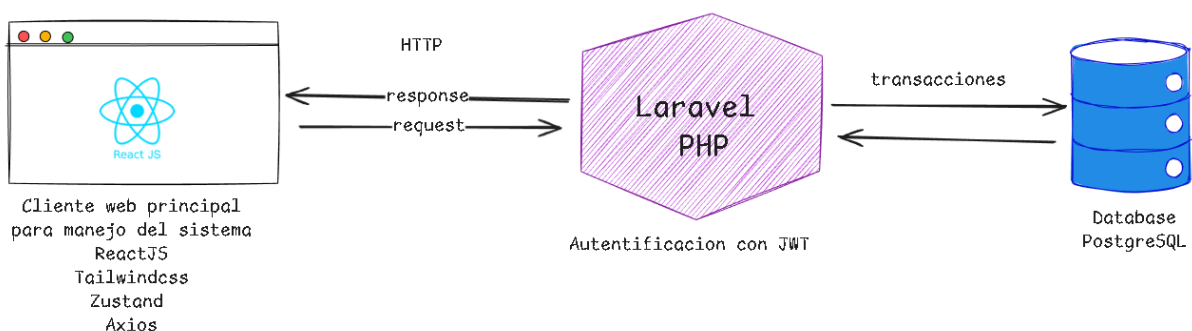
- **tarifa y horario:** Permite saber el precio del uso del espacio y el horario de uso de ese espacio.
- **servicio:** Tabla pivote para vincular servicios extra a una reserva.

4.8. Arquitectura del Sistema

Hemos desarrollado el sistema mediante una Arquitectura Desacoplada (Headless), una decisión estratégica de diseño que separa físicamente y lógicamente la capa de lógica de negocio (Backend) de la capa de interfaz de usuario o el cliente (Frontend). Para poder comunicar las dos capas que estamos usando se utilizaría una API RESTful estandarizada bajo el protocolo HTTP/HTTPS.

Este estilo arquitectónico, definido por (Fielding, 2000), impone restricciones como falta de estado (statelessness) y una interfaz uniforme, lo que mejora significativamente la escalabilidad, la portabilidad y la fiabilidad del sistema. Además, permite que equipos distintos trabajen en paralelo en cada capa sin bloquearse mutuamente y ayuda a facilitar la futura integración de nuevas interfaces que si se desean se pueden aplicar, así como lo sería un interfaz móvil dado el caso que en un futuro se requiera, obviamente sin modificar la lógica del servidor.

Figura 13. Arquitectura utilizada en el desarrollo del sistema.



Nota: Autoría propia. Explicación de la arquitectura del sistema.

4.8.1 Componentes Clave de la Arquitectura

1. **Capa de Presentación (Frontend – Cliente):** Se desarrollo utilizando el framework de React. El cliente se encarga exclusivamente de renderizar la interfaz de usuario y consumir los servicios de la API. Esta capa se desarrolló siguiendo el principio de Single Page Application (SPA). Según estudios recientes que hablan sobre el desarrollo web, dicen que las SPAs ofrecen una experiencia de usuario superior al reducir la latencia percibida y el consumo de ancho de banda ya que solo transmiten datos (JSON) en lugar de documentos HTML completos tras la carga inicial (Sangarsu, 2019).
2. **Capa de Servicios (Backend – Servidor):** Esta funciona como una API sólida que empleando el framework de Laravel, recibe solicitudes HTTP, valida la información, ejecuta la lógica comercial y devuelve respuestas en formato JSON.
3. **Capa de Persistencia (Base de Datos):** Eloquent ORM se emplea para la abstracción de datos en PostgreSQL, que guarda los datos de los usuarios de forma estructurada.

4.9. Desarrollo del Backend

Hemos estado hablando del backend durante todo el documento y el proyecto, este de aquí se encarga del núcleo lógico y funcional del sistema. Se desarrolló utilizando el framework de Laravel, aprovechando su arquitectura basada en el patrón MVC (Modelo-Vista-Controlador). Pero al estar hablando de API, la “Vista” es la representación JSON de los datos, el patrón MVC para poder separar de mejor manera varias partes del código cuándo se trata de preocupaciones, facilitando así el mantenimiento y la testabilidad del código (Pressman R. S., Ingeniería de Software: Un enfoque práctico (9.^a ed.). McGraw-Hill Education., 2020)

Entorno de Desarrollo y Stack Tecnológico:

- **Lenguaje:** PHP 8.2
- **Framework:** Laravel 10
- **Base de Datos:** PostgreSQL 15

- **Seguridad:** Laravel Sanctum para autenticación basada en tokens.

Estructura de la API y Controladores:

La lógica del sistema web que se encapsula en controladores específicos:

- **BookingController:** Se ocupa de todo el flujo de las reservas (crear, listar, actualizar)
- **EspacioController:** Gestiona el espacio físico (CRUD de espacios)
- **AuthController:** Se ocupa de toda la autenticación mediante Laravel Sanctum,

Implementación de Funcionalidades Esenciales:

- **Validación de Disponibilidad (Algoritmo de Negocio):** Utilizamos un método Scope reutilizable en el modelo de booking para poder verificar matemáticamente si una nueva fecha de solicitudes se intercepta con cualquier reserva existente confirmada.

Figura 14. Método encargado del manejo y validación de fechas.

```
0 references | 0 overrides
public function validarDisponibilidad($fecha_inicio, $fecha_fin, $espacio_id): bool
{
    // obtener el horario del espacio, filtrado por su activo
    $horario = Espacio::find(id: $espacio_id)→load(relations: 'horarios')→horarios→where
    (column: 'activo', operator: true);

    // formatear las fechas a Y-m-d H:i:s
    $fecha_inicio = $fecha_inicio→format('Y-m-d H:i:s');
    $fecha_fin = $fecha_fin→format('Y-m-d H:i:s');

    // recorrer los horarios
    foreach ($horario as $h) {
        // verificar si la fecha de inicio y fin estan dentro del rango del horario
        if ($fecha_inicio ≥ $h→fecha_inicio && $fecha_fin ≤ $h→fecha_fin) {
            // si cumple con el horario, retornar true
            return true;
        }
    }

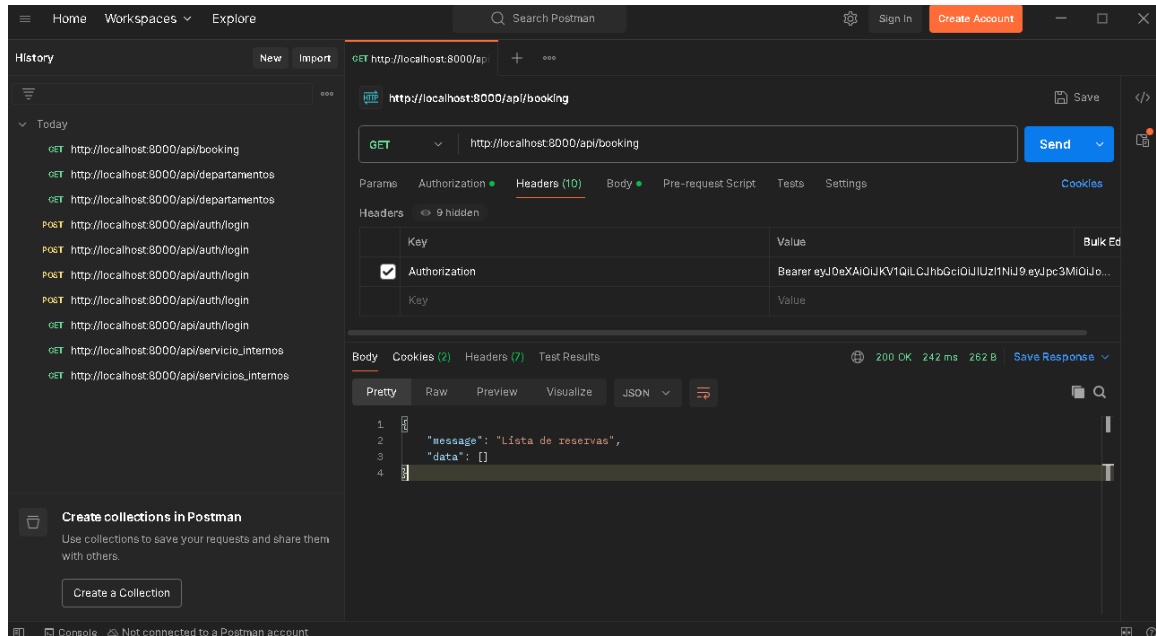
    // si no cumple con ningún horario, retornar false
    return false;
}
```

Nota: El siguiente método corresponde al controlador de reservas y valida la disponibilidad de fechas antes de mandar la solicitud.

Pruebas de la API:

Con el fin de probar que nuestro sistema web sea confiable, llevamos a cabo evaluaciones de black box con Postman mediante peticiones para confirmar estados HTTP.

Figura 15. Prueba de endpoint de listado de reservas con Postman.



Nota: Se listan las reservas realizadas a través del sistema, las cuales se consultan por el api también desde el cliente web, esta es una demostración en Postman.

Tabla 5. Endpoints utilizados.

Tabla de Endpoints

Método	URL	Nombre Ruta	Acción
POST	api/auth/login	<i>generated</i>	<u>AuthController</u> <u>@login</u>
POST	api/auth/login-microsoft	<i>generated</i>	<u>AuthController</u> <u>@loginMicroso</u> <u>ft</u>
POST	api/auth/logout	<i>generated</i>	<u>AuthController</u> <u>@logout</u>

POST	api/auth/register	<i>generated</i>	<u>AuthController</u> <u>@register</u>
GET	api/booking	booking.index	Api\BookingCo ntroller@index
GET	api/booking	booking.store	Api\BookingCo ntroller@store
GET	api/booking/{booking}	booking.show	Api\BookingCo ntroller@show

GET	api/booking/{booking}	booking.update	Api\BookingController@update
GET	api/booking/{booking}	booking.destroy	Api\BookingController@destroy
GET	api/booking-estados	<i>generated</i>	Api\EstadoBookingController@index
GET	api/booking/espacios_por_departamento/{id}	<i>generated</i>	Api\BookingController@solicitudesPorDepartamentosDeEspacios
GET	api/booking/mis-reservas/{id}	<i>generated</i>	Api\BookingController@misReservas

GET	api/booking/mis-reservas/aceptadas/{id}	<i>generated</i>	Api\BookingController@misReservasAceptas
GET	api/booking/mis-reservas/pendientes/{id}	<i>generated</i>	Api\BookingController@misReservasPendientes
GET	api/booking/mis-reservas/rechazadas/{id}	<i>generated</i>	Api\BookingController@misReservasRechazadas
GET	api/booking/{id}/aprove	<i>generated</i>	Api\BookingController@setEstadoAprovado
GET	api/booking/{id}/cancel	<i>generated</i>	Api\BookingController@setEstadoCancelado

GET	api/booking/{id}/comprobante-pago	<i>generated</i>	Api\BookingController@verComprobantePago
GET	api/booking/{id}/enviar-pago	<i>generated</i>	Api\BookingController@enviarPago
GET	api/booking/{id}/paid	<i>generated</i>	Api\BookingController@setEstadoPagoRealizado
GET	api/booking/{id}/reject	<i>generated</i>	Api\BookingController@setEstadoRechazado
GET	api/booking/{id}/request-payment	<i>generated</i>	Api\BookingController@setEstadoPagoPendiente

GET	api/booking/{id}/viewed	<i>generated</i>	Api\BookingController@setEstadoRevisado
POST	api/espacios	espacios.index	Api\EspacioController@index
GET	api/espacios/{espacio}	espacios.store	Api\EspacioController@store
PUT/PATCH	api/espacios/{espacio}	espacios.show	Api\EspacioController@show
DELETE	api/espacios/{espacio}	espacios.update	Api\EspacioController@update

GET	api/espacios/all-data	espacios.destroy	Api\EspacioController@destroy
GET	api/espacios/departamento/{id}	<i>generated</i>	Api\EspacioController@allWithAllRelations
POST	api/espacios/disponible/{id}	<i>generated</i>	Api\EspacioController@espaciosPorDepartamento
POST	api/espacios/{id}/asignar-mobiliarios	<i>generated</i>	Api\EspacioController@disponible
Resource	api/estados_espacios	<i>generated</i>	Api\EspacioController@asignarMobiliarios

Resource	api/categoria_espacios	estados_espacios.*	Api\EstadoEspacioController (CRUD)
Resource	api/mobiliarios	categoria_espacios.*	Api\CategoriaEspacioController (CRUD)
GET	api/mobiliarios/all-data	mobiliarios.*	Api\MobiliarioController (CRUD)
Resource	api/categoria_mobiliarios	<i>generated</i>	Api\MobiliarioController@all WithAllRelations
Resource	api/departamentos	departamentos.*	Api\DepartamentoController (CRUD)

GET	api/departamentos/all-data	<i>generated</i>	Api\DepartamentoController@departamentosConDepartamentoPadre...
GET	api/departamentos/{id}/all-data	<i>generated</i>	Api\DepartamentoController@departamentoConDepartamentoPadre...
Resource	api/categoria_departamentos	categoria_departamentos.*	Api\CategoriaDepartamentoController (CRUD)
Resource	api/cargos	cargos.*	Api\CargoController (CRUD)
GET	api/cargos/{id}/usuarios	<i>generated</i>	Api\CargoController@usuariosConCargo

GET	api/cargos/{id}/usuarios-excluir	<i>generated</i>	Api\CargoController@usuariosExcluir
Resource	api/usuarios	usuarios.*	Api\UserController (CRUD)
GET	api/usuarios/all-data	<i>generated</i>	Api\UserController@usuariosConAll
GET	api/usuarios/{id}/all-data	<i>generated</i>	Api\UserController@usuarioConAll
GET	api/usuarios/{id}/rol	<i>generated</i>	Api\UserController@showRol

Resource	api/roles	roles.*	Api\RolesController (CRUD)
GET	api/roles/permisos	<i>generated</i>	Api\RolesController@showAllWithPermissions
GET	api/roles/{id}/permisos	<i>generated</i>	Api\RolesController@showWithPermissions
Resource	api/permisos	permisos.*	Api\PermisosController (CRUD)
GET	api/permisos/{id}/usuarios	<i>generated</i>	Api\UserController@showPermissions

Resource	api/horarios	horarios.*	Api\HorarioController (CRUD)
GET	sanctum/csrf-cookie	sanctum.csrf-cookie	CsrfCookieController@show
GET	/	<i>generated</i>	Closure (Ruta raíz)
GET	up	<i>generated</i>	Closure (Health check)

Nota: En esta tabla se detallan cada uno de los endpoints que tiene el producto.

4.10. Desarrollo del Frontend

Para el desarrollo del frontend se utilizó el framework de ReactJS 18, empleando Vite como herramienta de construcción. React se basa en el concepto de componentes y un DOM virtual, lo que nos permite optimizar bastante el renderizado y mejorar también el rendimiento de la aplicación en comparación a una manipulación directa del DOM

Estrategia de gestión del estado: Se utilizó Zustand para la gestión del estado global, permitiendo compartir información crítica (como la sesión del usuario) de manera eficiente.

Interfaces Funcionales Implementadas:

1. Pantalla de Login y Seguridad

Figura 16. Formulario para el inicio de sesión.



The image shows a login form for a platform called 'Escenarios'. At the top, there is a logo consisting of a blue atom-like symbol and the word 'Escenarios'. Below the logo, the title 'Iniciar Sesión' is centered. A welcome message reads: 'Bienvenido a la plataforma de reservas de escenarios. Por favor inicia sesión para continuar.' The form contains two input fields: 'Correo Electrónico' with the placeholder 'Ingrese su correo electrónico' and 'Contraseña' with the placeholder 'Ingrese su contraseña'. Below these fields is a checkbox labeled 'Recordar mi sesión'. A large blue button labeled 'Iniciar Sesión' is positioned below the checkbox. A horizontal line separates this section from the next one, which is labeled 'SI ERES MIEMBRO DE LA COMUNIDAD UNIVERSITARIA'. Below this line is another large blue button labeled 'Iniciar Sesión con Microsoft365 ULEAM', preceded by a small Microsoft logo. At the bottom, there is a link that says 'Si no tienes una cuenta' followed by the word 'Regístrate'.

Nota: En la pantalla de inicio de sesión existe la posibilidad de iniciar sesión también con la cuenta institucional que claramente pertenecerá a un miembro de la comunidad universitaria.

2. Panel de Gestión de Espacios (Dashboard Admin)

Figura 17. Panel de vista de los espacios con los que cuenta la institución.

Gestión de Espacios
Administra espacios y salas disponibles en el sistema

Actualizar Agregar Espacio Gestionar Categorías

Buscar espacios

Buscar por nombre

Mostrando 0 espacios

NOMBRE	DESCRIPCIÓN	CAPACIDAD	ACCIONES
No hay espacios disponibles. Para poder agregar espacios, haz clic en el siguiente botón:			

+ Agregar un nuevo espacio

Nota: La vista comprende la gestión de los espacios los cuales son administrables desde este mismo, son estos espacios los que se solicitaran para alquiler.

Figura 18. Vista de un espacio detallado con precio, mapa y equipamiento.



4. Administración de espacios.

Figura 19. Dashboard para la administración de todos los espacios.

Gestión de Espacios
Administra espacios y salas disponibles en el sistema

Actualizar Gestionar Categorías Agregar Espacio

Buscar espacios

Buscar por nombre

Mostrando 2 espacios

NOMBRE	CATEGORÍA	DEPARTAMENTO	CAPACIDAD	HORARIO DE ATENCIÓN	ESTADO
Auditorio Columbina	Auditorios	Universidad Laica Eloy Alfaro de Manabí	254	Horario furina 10:00:00 - 20:00:00 - \$30/hora 19 de enero de 2026 - 19 de febrero de 2026 Por entrar	Disponible
Salon Columbina 2	Salones	Universidad Laica Eloy Alfaro de Manabí	255	Horario Yae Miko 08:00:00 - 15:00:00 - \$450/hora Sin limite de fechas Sin limite	En mantenimiento

Nota: Autoría propia.

5. Formulario para agregar nuevos espacios.

Figura 20. Formulario de crear nuevo espacio.

Crear Nuevo Espacio

Estado del espacio

Disponible

Información: Por primera vez, el espacio estará en el estado de "Disponible". Podrás cambiar el estado después de crear el espacio.

Categoría del espacio

Seleccionar una categoría

Departamento

Seleccionar un departamento

Nombre del espacio

Ingrese el nombre del espacio

Capacidad

Ingrese la capacidad del espacio

Descripción

Ingrese una descripción del espacio

Metros Cuadrados

Ingrese los metros cuadrados del espacio

Altura (m)

Ingrese la altura del espacio

Cancelar Crear Espacio

Nota: Autoría Propia.

6. Gestión de Usuarios.

Figura 21. Dashboard para la gestión de usuarios.

Gestión de Usuarios
Administra usuarios, roles y permisos del sistema

Actualizar Agregar Usuario

Buscar usuarios
Buscar por nombre, email o cédula... Limpiar filtros

Filtrar por rol: Todos los roles
Filtrar por estado: Todos los estados

Mostrando 8 usuarios

NOMBRE	EMAIL	ROL	DEPARTAMENTO	CARGO	FECHA DE CREACIÓN
Billie Eilish Pirate O'Connell CEDULA - 712963407162383	haleyscommet@email.com 712963407162383	administrador third	Universidad Laica Eloy Alfaro de Manabí ULEAM	Alisa Mikhailovna Kujou Cargo Alisa	11 de enero de 20
Kamisato Ayaka CEDULA - 600000003	ayaka.kamisato@email.com 600000003	administrador third	Universidad Laica Eloy Alfaro de Manabí ULEAM	Administrador	11 de enero de 20
Sangonomiya Kokomi CEDULA - 5357659312463448	sangonomiya.kokomi@email.com 5357659312463448	administrador third	Universidad Laica Eloy Alfaro de Manabí ULEAM	Alisa Mikhailovna Kujou Cargo Alisa	11 de enero de 20
Shogun Raiden CEDULA - 826329167282326	raiden.shogun@email.com 826329167282326	administrador	Universidad Laica Eloy Alfaro de Manabí ULEAM	Administrador	11 de enero de 20
Vegetta 777 CEDULA - 1111112222	vegetta@test.com 1111112222	rol usuario	Facultad de Psicología FAPS	Decano/a Decano/a	15 de enero de 20
CEDEÑO MOSQUERA JOSTIN ABRAHAM CEDULA - 1350797955	e1350797955@live.uleam.edu.ec 1350797955	Admin	Universidad Laica Eloy Alfaro de Manabí ULEAM	Estudiante	12 de enero de 20

Nota: Autoría Propia.

7. Gestión de Roles.

Figura 22. Dashboard para la gestión de roles.

Gestión de Roles
Administra roles, permisos y accesos del sistema

Actualizar Agregar Rol

Buscar departamentos
Buscar por nombre

Mostrando 7 departamentos

NOMBRE	CODIGO DEL ROL	FECHA DE CREACIÓN	ULTIMA ACTUALIZACIÓN	ACCIONES
delete_usuario	delete_usuario	9 de enero de 2026	9 de enero de 2026	🔍 ✎ 🚫
rol usuario	rol usuario	9 de enero de 2026	9 de enero de 2026	🔍 ✎ 🚫
rol admin	rol admin	9 de enero de 2026	9 de enero de 2026	🔍 ✎ 🚫
administrador	administrador	11 de enero de 2026	11 de enero de 2026	🔍 ✎ 🚫
administrador third	admin-third	11 de enero de 2026	11 de enero de 2026	🔍 ✎ 🚫
SUPER USER	super-user	12 de enero de 2026	12 de enero de 2026	🔍 ✎ 🚫
Admin	admin	12 de enero de 2026	12 de enero de 2026	🔍 ✎ 🚫

Nota: Autoría Propia.

4.11. Conclusiones del Capítulo

El desarrollo exitoso de un sistema web que es funcional, robusto y en línea con las normas actuales de la industria se demuestra en la propuesta técnica entregada. La integridad y la seguridad de la información se aseguran cuando se lleva a cabo de manera fiel el diseño de base de datos relacional y el control de acceso RBAC (R. S. Sandhu, 1996). La implementación de servicios RESTful (Fielding, 2000) y la arquitectura desacoplada React-Laravel han resultado ser decisiones estratégicas adecuadas, ya que se ha conseguido una solución que puede ampliarse y mantenerse con el tiempo, lo que permite modernizar la administración de los recursos físicos de la ULEAM.

Capítulo V: Evaluación de Resultados

5.1. Introducción.

Al entrar en este capítulo estamos entrando en una fase crítica de la tesis, centrada en la validación objetiva de los avances alcanzados en el desarrollo del “Sistema para la Gestión y Administración del Uso de Escenarios en la ULEAM, Matriz Manta”. Al haber finalizado las primeras iteraciones de la metodología ágil, Programación Extrema (XP), continuamos a evaluar si el producto tecnológico cumple con los requisitos funcionales y técnicos establecidos. El alcance de esta evaluación contempla la operatividad de los módulos de seguridad, la administración de espacios físicos y la lógica transaccional inicial, pudiendo demostrar que nuestro sistema web no solo es una propuesta teórica, sino una herramienta completa y funcional que se alinea con los objetivos de modernización institucional de la universidad.

En esta fase nuestro software cuenta con una API backend basada en Laravel y un cliente web en React en el cual estamos implementando navegación y módulos administrativos/operativos. Documentamos los resultados mediante evidencias técnicas (rutas implementadas, componentes, servicios API, configuración de autenticación) y pruebas funcionales o de integración.

5.2. Descripción del proceso de evaluación.

Para poder determinar la validez de los resultados obtenidos hasta esta etapa, se estableció un proceso de evaluación basado en la verificación de objetivos y desempeño técnico.

5.2.1 Criterios de Evaluación

La funcionalidad de nuestro sistema se evaluó mediante pruebas de creación, consulta y validación de datos usando Postman, los criterios que elegimos fueron estos:

Tabla 6. Resultados parciales.

Matriz de evaluación de resultados parciales.

Criterio	Indicador Específico	Herramienta	Resultado Esperado
Funcionalidad	Las operaciones CRUD se ejecutan correctamente (Crear, Leer, Actualizar, Eliminar)	Interfaz de usuario y verificación en la base de datos.	Los registros tanto de usuarios o roles se crean y modifican en la base de datos sin errores.
Seguridad	Validamos los accesos a rutas protegidas y el inicio de sesión.	Postman y LocalStorage.	El sistema no permite el acceso si no hay token y si, si existe el token, permite el acceso.
Integración	Comunicación asíncrona entre el cliente y el servidor.	Devtools y Axios.	Respuestas HTTP con código 200 OK y estructura JSON.

Nota: Estos 3 criterios se tiene que cumplir para tener en cuenta de que la fase de evaluación está completada.

5.2.2 Relación con los objetivos del proyecto

Cada resultado que hemos obtenido con este proyecto de tesis y más con los resultados de este capítulo, guarda una relación directa con los objetivos específicos de la tesis. Por ejemplo, el desarrollo del módulo de seguridad cumple con el requerimiento de establecer un control de acceso riguroso, mientras que la administración de roles se necesita debido a jerarquizar las funciones administrativas dentro de la institución para garantizar la transparencia.

5.3. Presentación de resultados

En este punto vamos a detallar las funcionalidades operativas y las evidencias técnicas del sistema.

5.3.1 Resultados técnicos y funcionalidades implementadas

La siguiente tabla detalla las pruebas realizadas a los módulos creados dentro del sistema, Usuarios, Roles, Espacios e Interfaz de Usuarios.

Tabla 7. Matriz de pruebas.

Matriz de sistematización de resultados del sistema.

Función	Descripción de la Prueba	Resultado Esperado	Resultado Obtenido	Evidencia
Gestión de Espacios.	Petición POST con el body de JSON para agregar un nuevo espacio.	El servidor debe devolver el código 200 OK y también el espacio creado.	Exitoso.	Figura XVIII y XIX.

Consulta de Espacios.	Petición GET para poder obtener todos los espacios que existen en la base de datos.	Devolución de un array con todos los espacios.	Exitoso.	Figura XX.
Gestión de Usuarios.	Registro de un nuevo usuario mediante un POST.	Creación del usuario en la base de datos y nos devuelve exitoso.	Exitoso.	Figura XX1 y XXII.
Consulta de Usuarios.	Conseguimos todos los usuarios mediante un GET.	Devolución de una lista en un JSON de todos los usuarios.	Exitoso.	Figura XXIII.
Gestión de Roles.	Asignamos los roles y también creamos los mismos.	Control de acceso mediante la persistencia del rol.	Exitoso.	Figura XXIV y XXV.

Consulta de Roles.	Petición GET para visualizar todos los roles creados en la base de datos.	Devolución de una lista en JSON de todos los roles.	Exitoso.	Figura XXVI.
Interfaz de Acceso.	Intento de inicio de sesión.	Acceso al sistema y al proceso completo de reserva.	Exitoso.	Figura XXVII.
Interfaz de Home.	Intento de navegación al Home.	Acceso al renderizado principal del producto.	Exitoso.	Figura XXVIII.
Interfaz de Dashboard.	Interfaz visual para el administrador.	Despliegue del dashboard para el administrador.	Exitoso.	Figura XXIX.

Interfaz de Booking.	Interfaz visual para poder hacer la solicitud del espacio.	Despliegue del formulario de booking o reserva.	Exitoso.	Figura XXX.
-----------------------------	--	---	----------	-------------

Laravel Sanctum.	Verificación en devtools de los tokens.	Token guardado en localStorage.	Exitoso.	Figura XXXI.
-------------------------	---	---------------------------------	----------	--------------

Nota: Matriz para las pruebas que se hicieron respecto a los resultados técnicos.

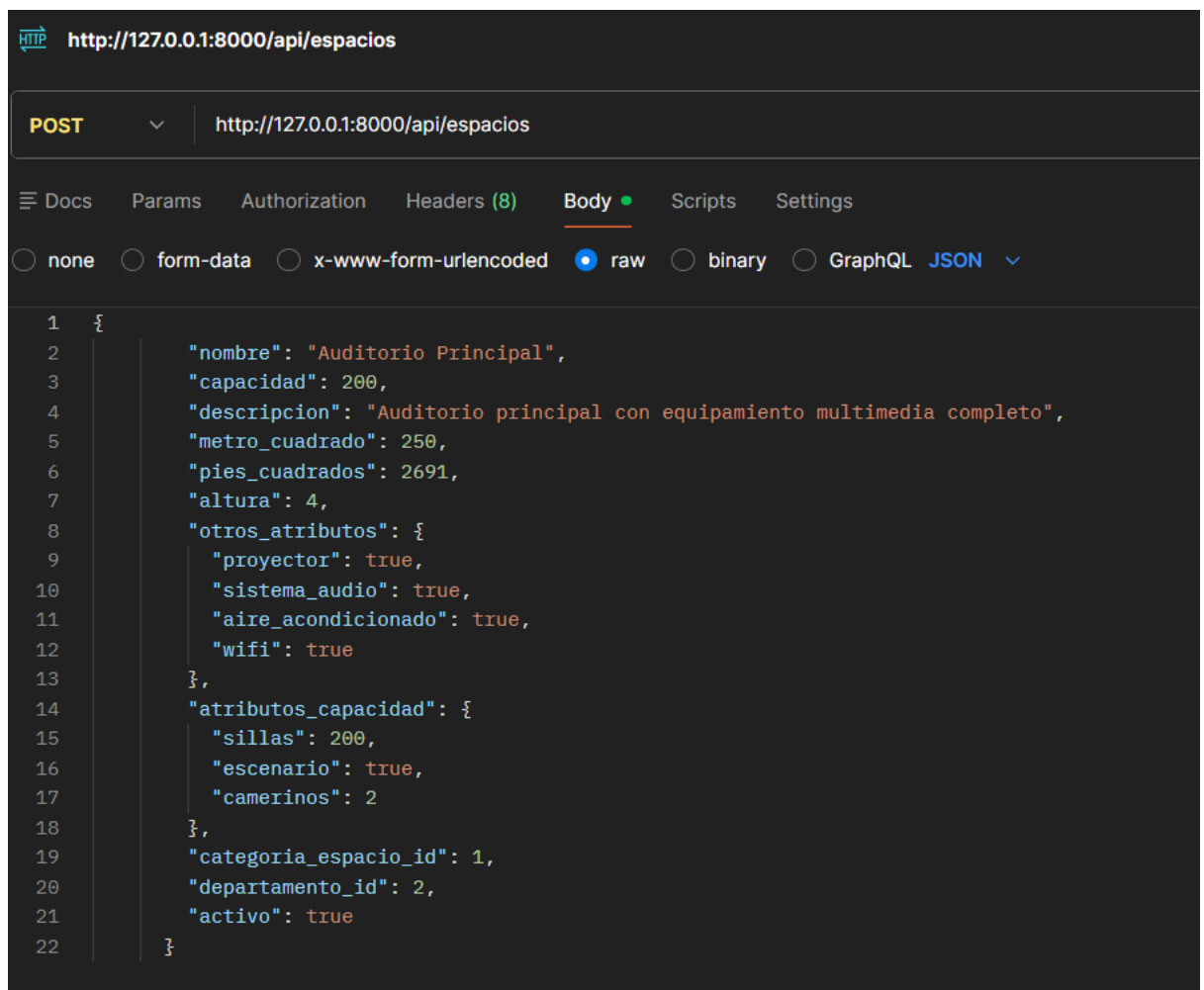
5.3.2 Evidencias de pruebas realizadas

Se documentaron pruebas de cajas negra para asegurar que el sistema se comporte según lo esperado.

Pruebas de Integración (API):

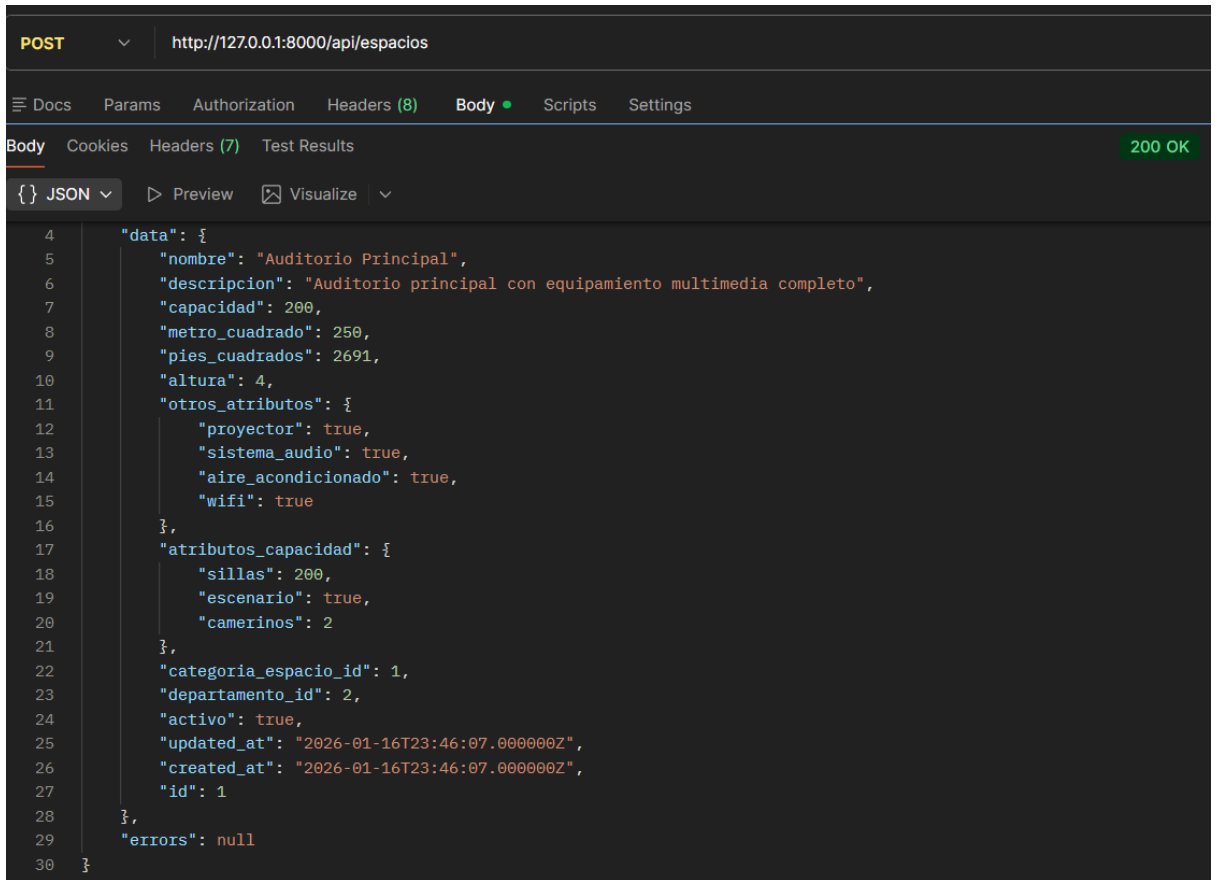
POST Espacios

Figura 23. Body Json para hacer el POST de espacios



Nota: Se escribe el JSON con el cuál se hace el POST del espacio dentro del backend.

Figura 24. Respuesta correcta del POST de espacios.



```
POST http://127.0.0.1:8000/api/espacios

Body Cookies Headers (7) Test Results 200 OK

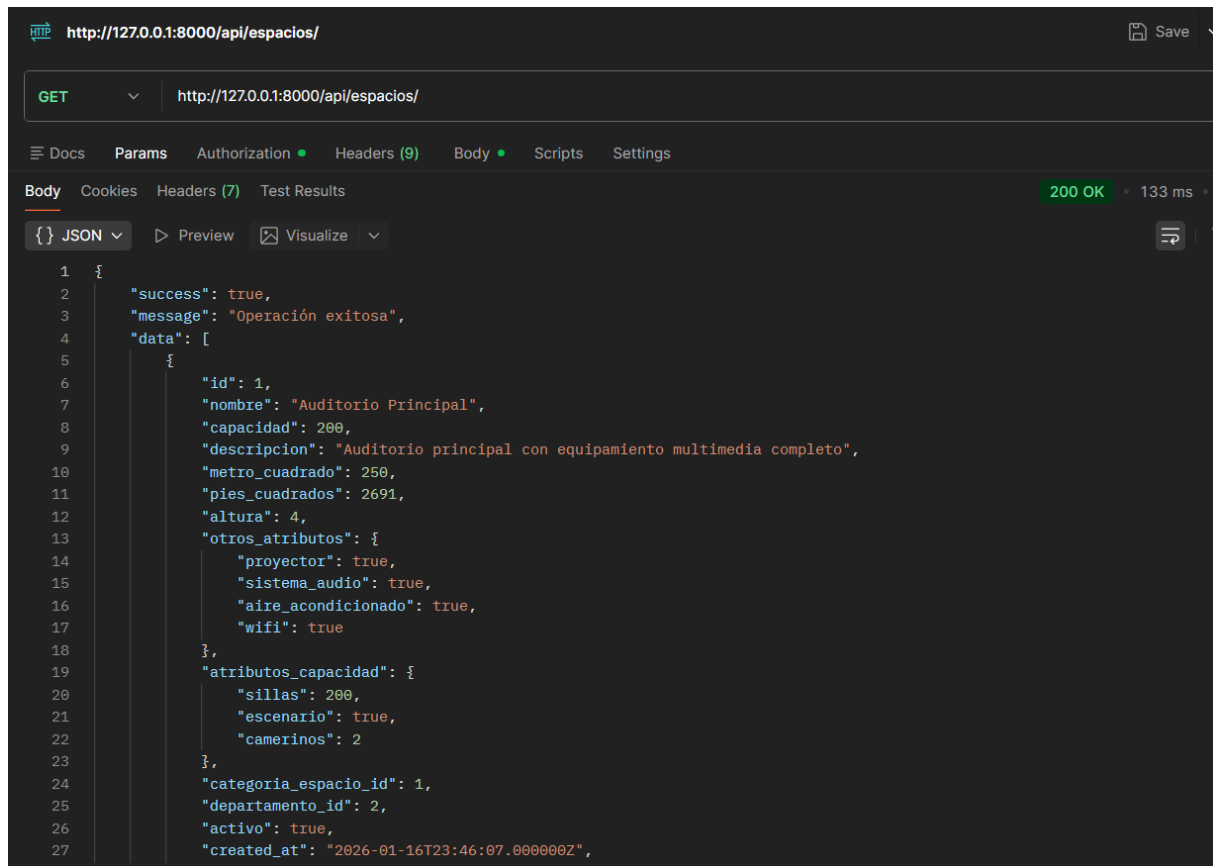
JSON Preview Visualize

4  "data": {
5      "nombre": "Auditorio Principal",
6      "descripcion": "Auditorio principal con equipamiento multimedia completo",
7      "capacidad": 200,
8      "metro_cuadrado": 250,
9      "pies_cuadrados": 2691,
10     "altura": 4,
11     "otros_atributos": {
12         "proyector": true,
13         "sistema_audio": true,
14         "aire_acondicionado": true,
15         "wifi": true
16     },
17     "atributos_capacidad": {
18         "sillas": 200,
19         "escenario": true,
20         "camerinos": 2
21     },
22     "categoria_espacio_id": 1,
23     "departamento_id": 2,
24     "activo": true,
25     "updated_at": "2026-01-16T23:46:07.000000Z",
26     "created_at": "2026-01-16T23:46:07.000000Z",
27     "id": 1
28 },
29 "errors": null
30 }
```

Nota: Se puede visualizar el código 200 OK que nos afirma que el POST se hizo de manera correcta dentro del backend.

GET Espacios

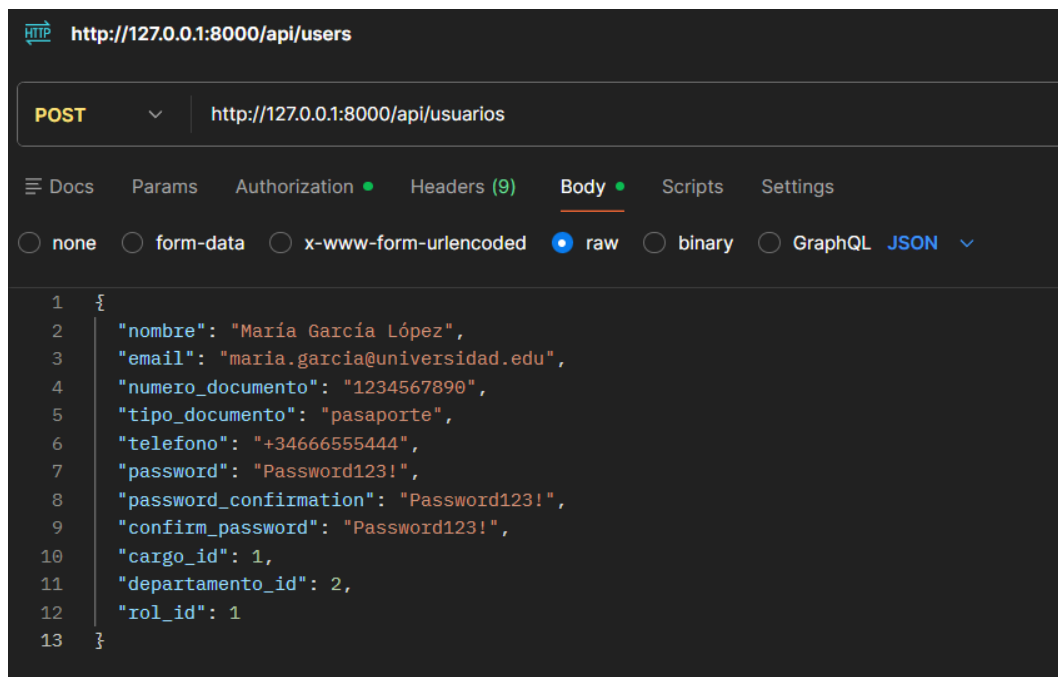
Figura 25. Devolución de la respuesta usando GET en espacios.



Nota: Podemos visualizar que al utilizar el GET en espacios, nos devuelve el espacio que recién agregamos con el POST.

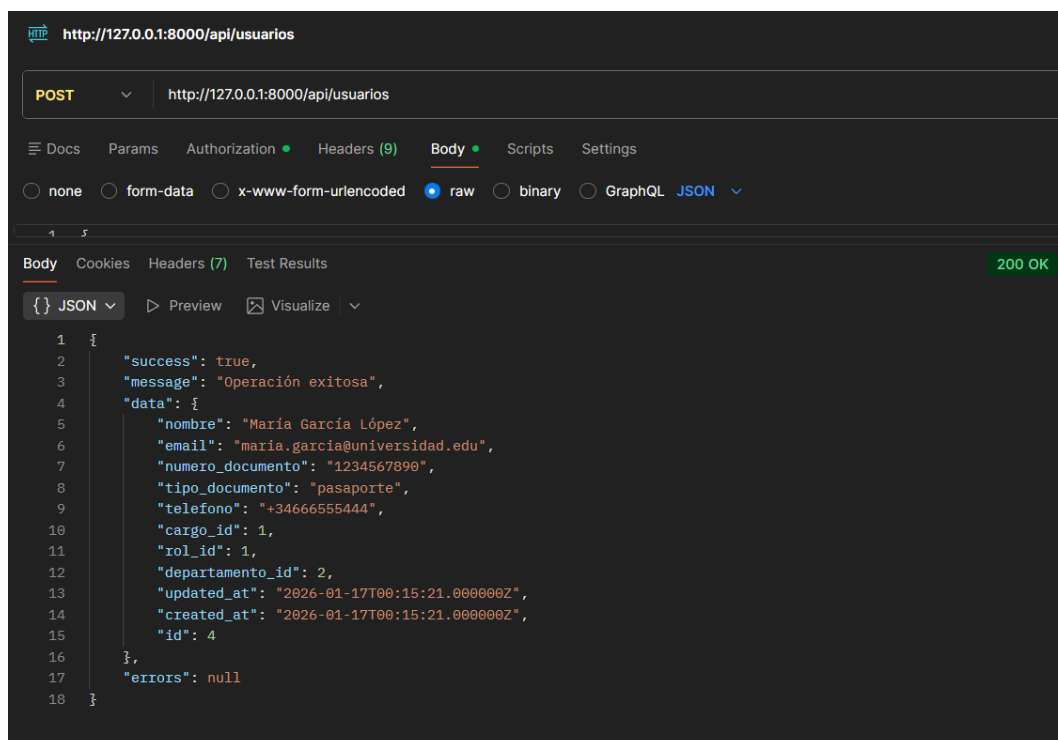
POST Users

Figura 26. Body JSON para el post de usuarios.



Nota: Podemos visualizar el Body JSON para hacer el POST dentro de users.

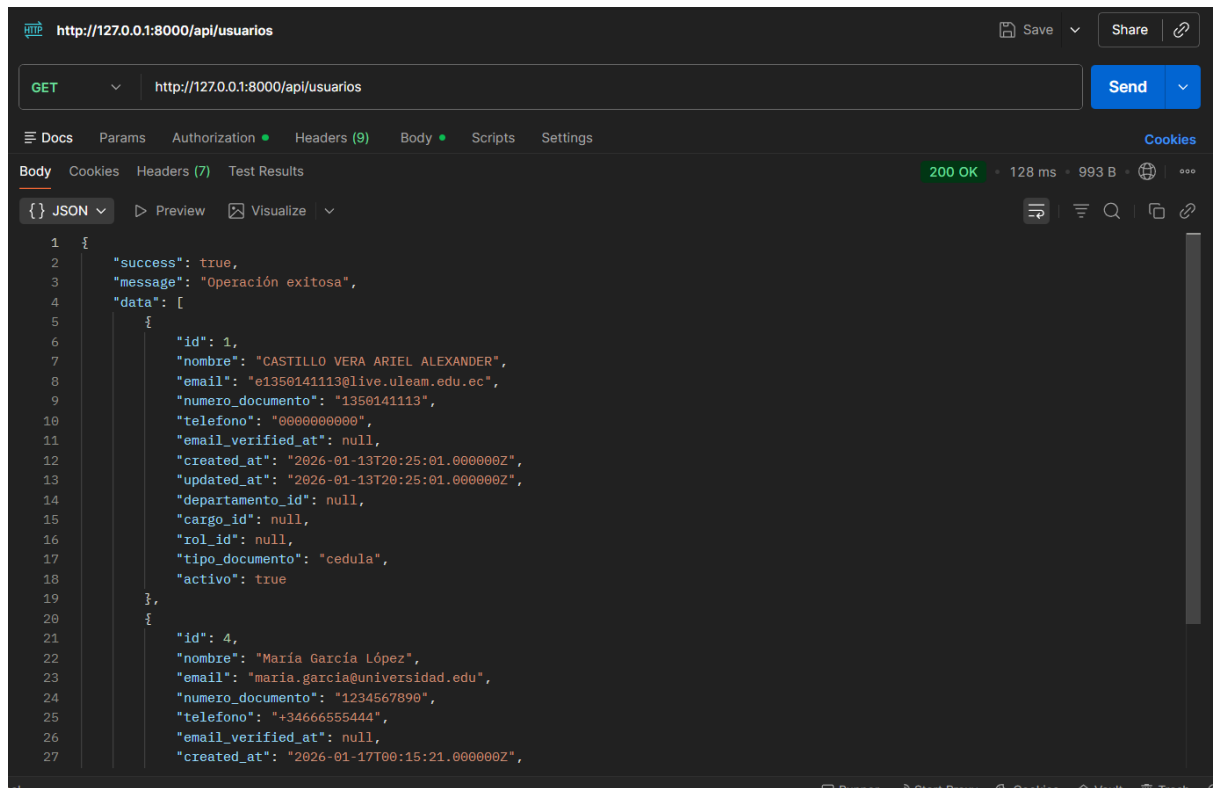
Figura 27. Respuesta correcta del POST usuarios.



Nota: Podemos visualizar que nos devuelve correctamente el POST de usuarios.

GET Usuarios

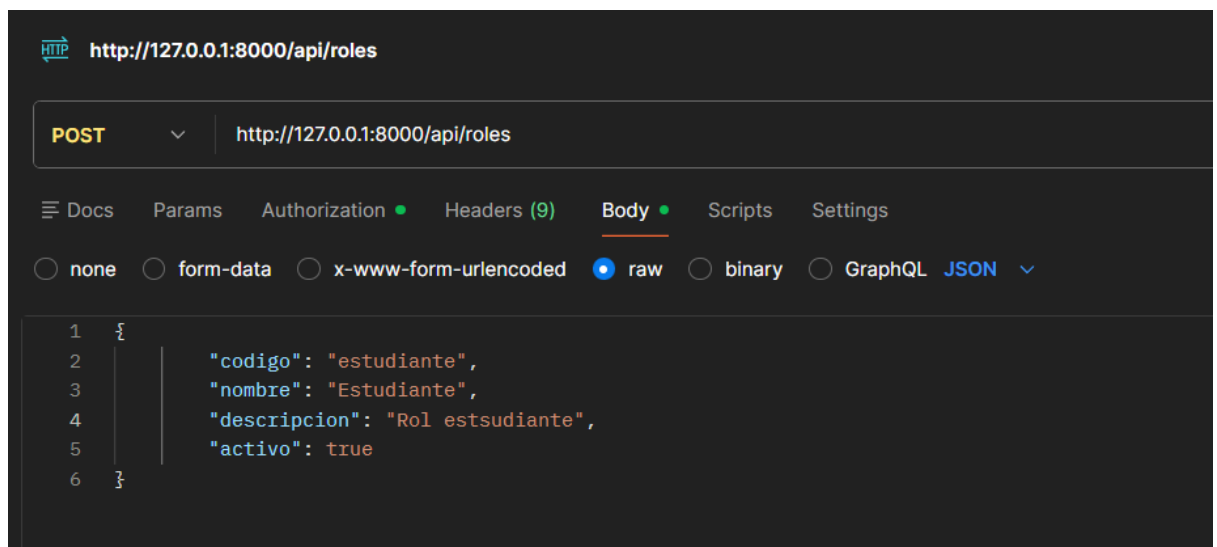
Figura 28. Respuesta devuelta de todos los usuarios.



Nota: Podemos observar que nos devuelve correctamente todos los usuarios.

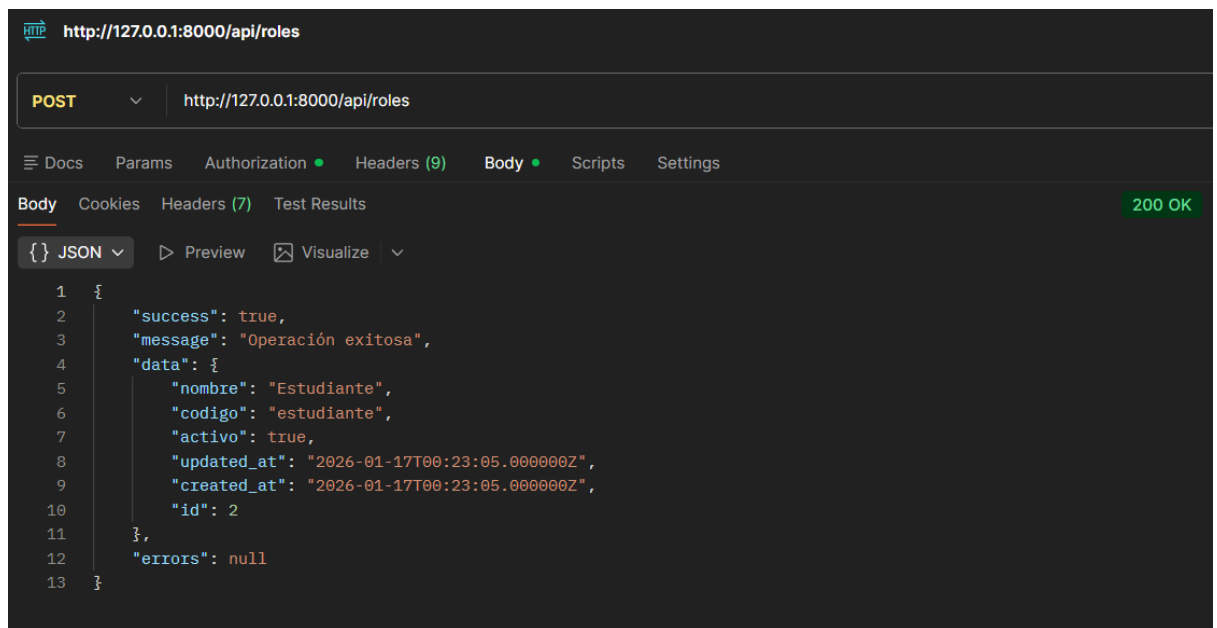
POST Roles

Figura 29. Body JSON para el POST de roles.



Nota: Tenemos en la figura el body JSON para el post de roles.

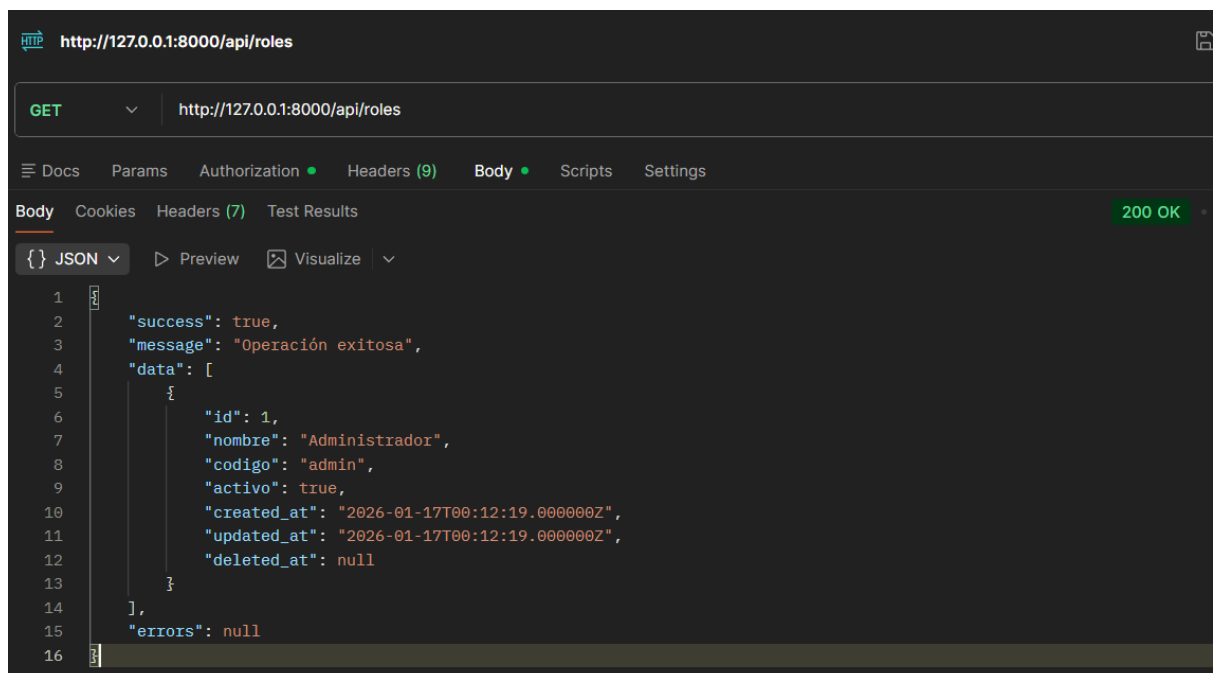
Figura 30. Respuesta correcta del POST de roles.



Nota: Podemos observar que nos devolvió correctamente la respuesta al hacer el POST.

GET Roles

Figura 31. GET de todos los roles.

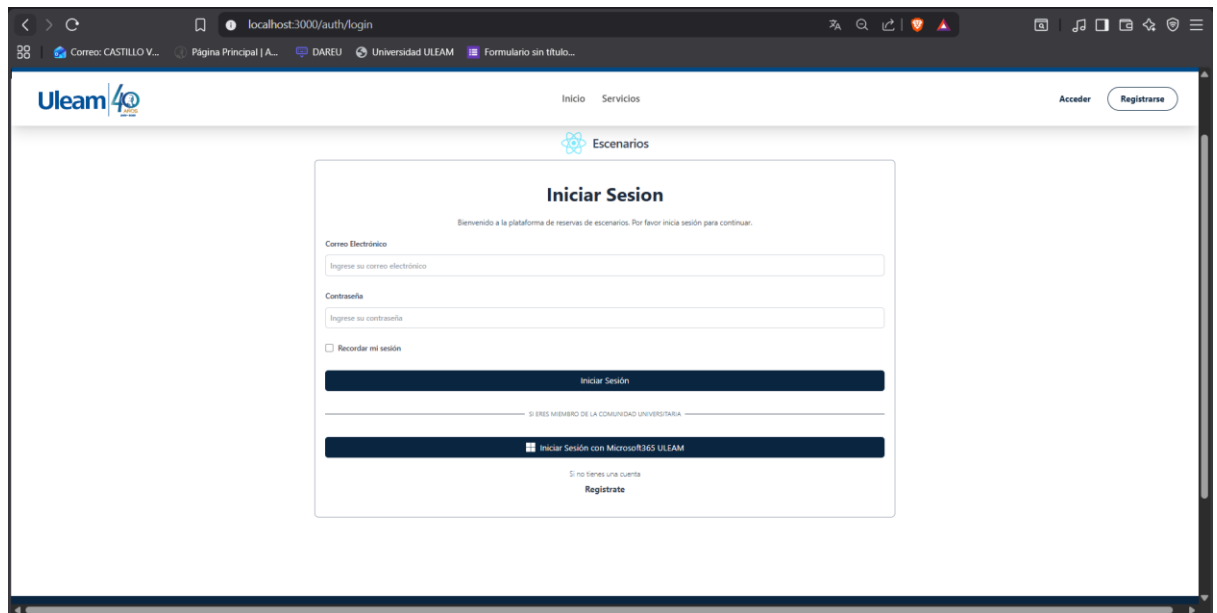


Nota: Tenemos la respuesta devuelta del get de roles dónde nos da el JSON del POST que hicimos en roles.

Pruebas Funcionales (Frontend):

LOGIN:

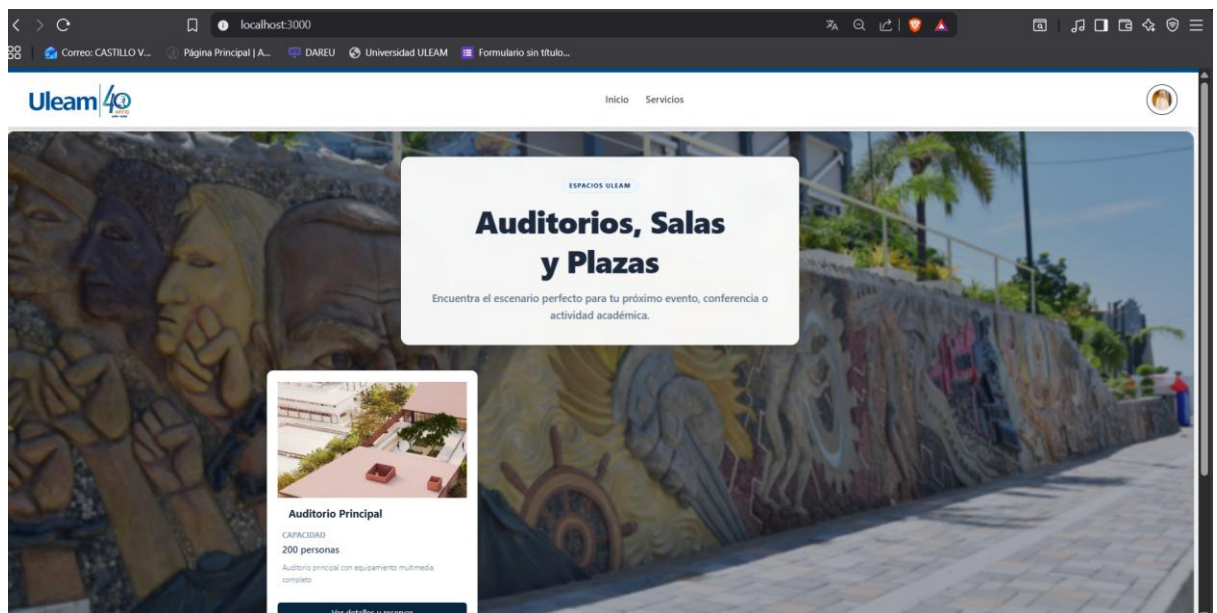
Figura 32. Pantalla de Login.



Nota: Podemos observar la página de login que tenemos para el sistema web.

HOME – CORELAYOUT:

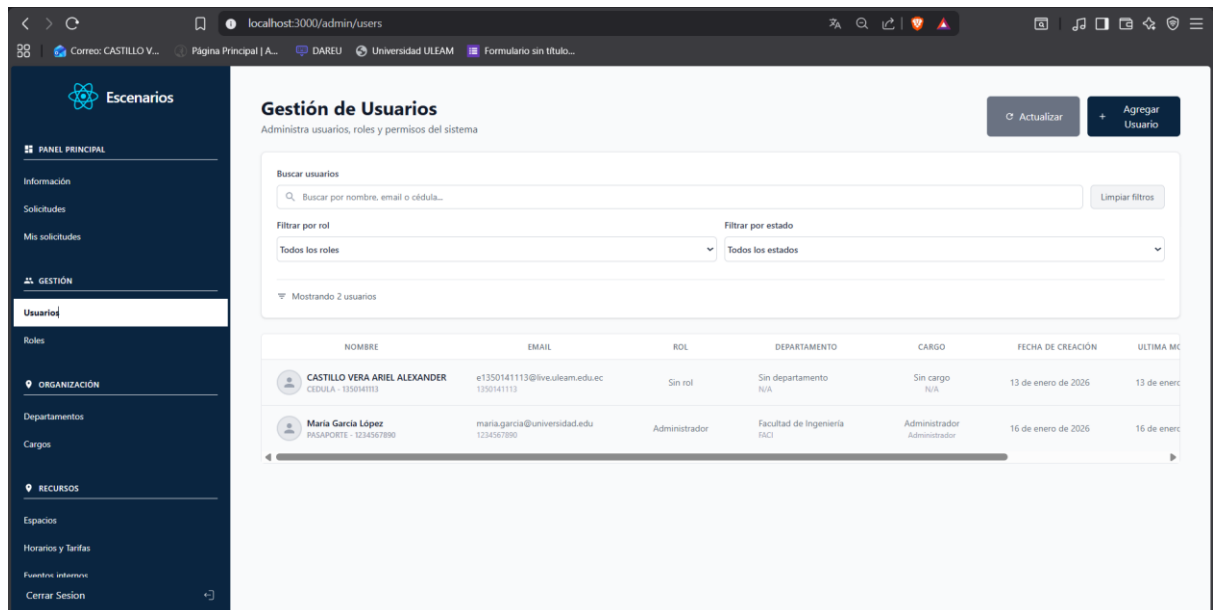
Figura 33. Pantalla Home



Nota: Podemos observar la página de home de nuestro sistema web.

DASHBOARD ADMINISTRATIVO:

Figura 34. Pantalla Dashboard Administrativo.



Nota: Podemos observar la página de dashboard administrativo, dónde encontramos usuarios, roles, espacios, servicios, etc.

FORMULARIO DE SOLICITUD

Figura 35. Interfaz de Formulario.

Generar solicitud

1 Información del Evento — 2 Contactos y Confirmación

Auditorio Columbina
Capacidad: 254 personas

Información del Evento

Nombre del evento
Ej: Conferencia de Tecnología 2026

Tipo de evento
Seleccionar tipo

Número de asistentes
Ej: 50

Descripción del evento
Describe brevemente el evento...

Fechas y Horarios

Horario del espacio: 10:00 - 20:00 | Vigencia: 19 ene 2026 - 19 feb 2026

Selecciona las fechas *

Nota: Formulario de solicitud para hacer el booking del espacio.

LARAVEL SANCTUM AUTHENTICATION

Figura 36. Token Personal

http://localhost:3000

Key	Value
APP_AUTH_TOKEN	eyJ0eXAiOiJV1QilCjhbGciOiJIUz11NiJ9.eyJpc3MiOiIodHRwOi8vMTI3LjAuMC4xOjgwMDAwYX8pL2F1dG9vbG9naW4tbWljcm9zb2Z0LiwiaWF0Ij0eX...
APP_AUTH_USER	{ "id": 1, "nombre": "CASTILLO VERA ARIEL ALEXANDER", "email": "e1350141113@live.uleam.edu.ec", "numero_documento": "1350141113", "telefono": "09950141113", "password": "1234567890" }
placesData	[{ "id": 1, "name": "Auditorio, Paraninfo Alfonso Aguilar Ruilova.", "title": "Auditorio, Paraninfo Alfonso Aguilar Ruilova", "subtitle": "Paraninfo Alfonso Aguilar Ruilova" }, ...]

Nota: Podemos observar que si tiene efectivamente autenticación el sistema web mediante el token personal.

5.4. Análisis e interpretación de resultados

Realizando un análisis crítico de esta versión de nuestro sistema web permite identificar que el sistema posee una base arquitectónica resiliente. El uso de Laravel nos ha hecho más simple la creación de una API escalable, mientras que React nos ofrece una reactividad superior en los formularios.

Como nuestra principal fortaleza al momento de usar Laravel, destaca el desacoplamiento total de las capas, lo que facilita el mantenimiento futuro. Como limitación, podemos observar que aún falta agregar la persistencia final de los formularios de reservar o bookings, los cuáles aún necesitan un pequeño arreglo en el frontend respecto a su diseño y que podamos visualizar, ya que de manera lógica en el backend ya está implementado. Este avance demuestra que el proyecto avanza según la metodología XP, priorizando la calidad de código sobre la cantidad de módulos incompletos.

Además, la implementación de una arquitectura desacoplada, utilizando tanto como Laravel para tener una API escalable y React para el cliente con sus interfaces dinámicas, resuelve plenamente el objetivo de utilizar tecnologías modernas, utilizar este enfoque ha dado al sistema de una base arquitectónica resiliente, mejorando la carga de datos a la base de datos y garantizando también su transparencia.

Los resultados obtenidos del módulo de autenticación cumplen con el segundo objetivo específico, mediante el uso de tokens con Laravel Sanctum y la gestión de roles, se ha garantizado un mecanismo de control y seguridad robusto. Esto nos genera un sistema donde

usuarios tanto internos como externos pueden entrar al sistema bajo protocolos estrictos de accesibilidad.

Construimos la lógica de negocio necesaria en el backend para el registro y revisión de solicitudes para el booking de los espacios, cumpliendo con el tercer objetivo específico de nuestra tesis.

5.5. Entrega del producto final

Se pone a disposición el acceso a los componentes desarrollados como evidencia técnica.

- Repositorio de GitHub Backend: <https://github.com/bocchii2/backendv2> (rama dev)
- Repositorio de GitHub Frontend: <https://github.com/bocchii2/client-scenarios-management> (rama dev)

Evidencias de pruebas finales

Figura 37. Reservación de espacios.

The screenshot displays the 'Generar solicitud' (Generate request) form for the Auditorio Columbina. The form is divided into two steps: 'Información del Evento' (Event Information) and 'Contactos y Confirmación' (Contacts and Confirmation). The first step is active, showing fields for event name, type, number of attendees, and description. The second step shows the event's location, capacity, and available dates.

Generar solicitud

1 Información del Evento — 2 Contactos y Confirmación

Auditorio Columbina
Capacidad: 498 personas [Salas de eventos](#)

Información del Evento

Nombre del evento
Auditorio Columbina

Tipo de evento
Ferias

Número de asistentes
Ej: 50

Descripción del evento
Descripción de eventos

Fechas y Horarios

Horario del espacio: 08:00 - 21:00 | Vigencia: 19 ene 2025 - 30 may 2026

Selecciona las fechas *

Nota: Apreciamos la primera parte de la solicitud para poder reservar un espacio.

Figura 38. Reservación de espacios 2

Generar solicitud

ene 2026 feb 2026

lun mar mié jue vie sáb dom lun mar mié jue vie sáb dom

29 30 31 1 2 3 4 26 27 28 29 30 31 1

5 6 7 8 9 10 11 2 3 4 5 6 7 8

12 13 14 15 16 17 18 9 10 11 12 13 14 15

19 20 21 22 23 24 25 16 17 18 19 20 21 22

26 27 28 29 30 31 23 24 25 26 27 28 1

28 29

Eventos Existentes en las Fechas Seleccionadas

El espacio tiene 1 evento(s) en este rango de fechas:

ariel 2
Horario Ocupado: 10:00 - 15:00

Tip: Puedes crear tu evento en un horario diferente que no se traslade con las reservas existentes.

Hora de inicio: 04:00 PM Hora de fin: 08:00 PM

Cancelar Siguiente

Nota: Podemos apreciar la segunda parte del formulario de reservación de espacios con los horarios.

Figura 39. Reservación de espacios 3.

Generar solicitud

Contacto principal: Columba Hiposelenia

Contacto responsable: Columba Hiposelenia

Resumen de la Solicitud

Espacio: Auditorio Columba

Evento: Auditorio Columba

Fechas: 27 ene 2026 - 28 ene 2026

Horario: 16:00 - 20:00

Costo estimado: \$96.00

Términos y Condiciones

☒ Acepto los términos y condiciones de uso del espacio

Ver términos y condiciones

Anterior Cancelar Enviar Solicitud

Nota: Podemos observar la última parte del formulario dónde se aprecia el resumen de la solicitud.

Figura 40. Revisión de solicitudes.

The screenshot displays the 'Escenarios' dashboard with a sidebar menu on the left containing sections like 'PANEL PRINCIPAL', 'GESTIÓN', 'ORGANIZACIÓN', and 'RECURSOS'. The main content area is titled 'Auditorio Columbina' and 'Solicitud #4', with a 'Pendiente' status badge. It features four panels: 'Información del Evento' (Event Information), 'Espacio Solicitado' (Requested Space), 'Progreso de la Solicitud' (Request Progress), and 'Información de Contacto' (Contact Information). The progress panel shows a timeline with 'Solicitud Enviada', 'En Revisión', and 'Decisión Final'. A 'Solicitud Pendiente' notification is also present.

Información del Evento	
Nombre del Evento	Auditorio Columbina
Descripción	Descripción de eventos
Tipo de Evento	Ferias
Fecha	28 de enero de 2026 - 29 de enero de 2026
Horario	16:00 - 20:00

Espacio Solicitado	
Nombre del Espacio	Auditorio Columbina
Departamento	Universidad Laica Eloy Alfaro de Manabí
Capacidad	498 personas
Área	120 m²
Dirección	N/A

Información Adicional	
Fecha de Solicitud	N/A
Revisado	No

Información de Contacto	
Contacto Principal	

Nota: En la figura se puede observar el dashboard para la revisión de solicitudes, con una pequeña trazabilidad.

Figura 41. Información general de las solicitudes.

The screenshot displays the 'Escenarios' dashboard with a sidebar menu on the left. The main content area is titled 'Banner Columbina 5 estrellas' and 'Solicitud #1', with an 'Aprobado' status badge. It features four panels: 'Información del Evento' (Event Information), 'Información del Espacio' (Space Information), 'Información del Solicitante' (Applicant Information), and 'Pago y Estado' (Payment and Status). The 'Pago y Estado' panel shows the current state as 'Aprobado' and includes a 'Requiere Pago' toggle.

Información del Evento	
Nombre del Evento	Banner Columbina 5 estrellas
Descripción	Banner Columbina 5 estrellas
Tipo de Evento	Conferencias
Fecha Inicio	23 de enero de 2026, 00:00
Fecha Fin	24 de enero de 2026, 00:00
Contacto Principal	CEDÑO MOSQUERA JOSTIN ABRAHAM
Contacto Responsable	CEDÑO MOSQUERA JOSTIN ABRAHAM

Información del Espacio	
Nombre del Espacio	Auditorio Columbina
Departamento	Universidad Laica Eloy Alfaro de Manabí
Capacidad	498 personas
Área	120 m²
Dirección	N/A
Descripción	Auditorio Columbina

Información del Solicitante	
Nombre	CEDÑO MOSQUERA JOSTIN ABRAHAM
Email	e1350797955@live.uileam.edu.ec
Tipo de Documento	Número de Documento
Cedula	1350797955
Teléfono	0000000000

Pago y Estado	
Estado Actual	Aprobado
Requiere Pago	☒
Pago Confirmado	✓ SI
Revisado	

Nota: En la figura vemos como se mostraría la información general de una solicitud, tanto como información del evento, del espacio, del solicitante además de pago y estado, si es que requiere.

Conclusiones

Con el análisis integral realizado en el desarrollo de la presente tesis, podemos concluir que esta gestión de los escenarios universitarios de la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta, presenta limitaciones significativas derivadas del uso de procesos manuales y descentralizados para el uso o alquiler de estas mismas, los cuales afectan de manera directa la eficiencia administrativa, tener transparencia institucional y la adecuada utilización de recursos físicos disponibles, debido a eso, estas deficiencias se manifiestan principalmente en la duplicación de reservas, la ausencia de trazabilidad histórica, la falta de información en tiempo real y las dificultades para una planificación estratégica efectiva.

Al estudiar nuestro marco teórico permitimos evidenciar que la transformación digital constituye un eje fundamental para el fortalecimiento de la gestión universitaria en el contexto actual. Estas experiencias documentadas a nivel internacionales, nacional y regional nos demuestran que tener una implementación de sistemas de información basados en aplicaciones web contribuye de manera significativa a la optimización de procesos administrativos, poder reducir el costo operativo y mejorar al momento de tomar decisiones importantes en la institución. En este sentido, la investigación confirma la adopción de tecnologías de la información no nos ayuda solamente de manera técnica, sino un cambio estructural orientado a la modernización organizacional.

De la misma manera, podemos concluir que las aplicaciones web, bajo una arquitectura cliente-servidor y con un enfoque desacoplado, se convierten en una solución tecnológica adecuada para el uso de las instituciones de educación superior que requieren accesibilidad y mantenibilidad eficiente. Al usar tecnologías modernas como React para la parte del cliente y Laravel como servidor y Postgresql como sistema de gestión de base de datos nos garantiza una aplicación robusta, segura y alineada con estándares actuales de desarrollo de software, permitiendo que se gestione de manera centralizada los escenarios universitarios.

Partiendo desde el visto de punto metodológico, esta investigación aplicada con enfoque descriptivo y tecnológico resultó pertinente para abordar la problemática identificada, ya que permitió analizar el estado actual de la gestión de escenarios y a partir de ellos, tener la propuesta de una solución correcta y que también sea viable. Tener la combinación de métodos teóricos y empíricos nos facilitó mucho la identificación de requerimientos funcionales y no funcionales, así pudimos asegurar que la propuesta responde a las necesidades reales de los usuarios internos y externos de la ULEAM.

Haber seleccionado la metodología de desarrollo ágil, Programación Extrema (XP), la consideramos adecuada para el contexto del proyecto, debido a todo su enfoque iterativo, la entrega continua de valor y la posibilidad de incorporar retroalimentación constante. Usar esta metodología favorece demasiado a la calidad del software, su adaptabilidad a poder tener cambios en los requerimientos y al desarrollo progresivo de una solución alineada con aquellos objetivos institucionales, de esa manera resulta especialmente relevante en proyectos académicos que tengan alcance definido pero susceptible de mejoras incrementales.

Pasando al ámbito legal y normativo, podemos concluir el desarrollar una aplicación web para la gestión de escenarios universitarios, se encuentra plenamente respaldado por la Constitución de la República del Ecuador, la Ley Orgánica de Educación Superior y la normativa institucional de la ULEAM. Esta propuesta contribuye a los cumplimientos de los propios principios de eficiencia, transparencia, igualdad de oportunidades, y adecuada administración de los recursos públicos, fortaleciendo la gobernanza universitaria y el control institucional.

Finalmente, concluimos que la propuesta de desarrollo de un sistema web para la administración y gestión de escenarios universitarios en la ULEAM es técnica, metodológica y legalmente viable. Su implementación permitirá a la universidad mejorar exponencialmente la planificación institucional, optimizar el uso de estos espacios físicos, reducir aquellos conflictos operativos que salgan y brindar una experiencia más eficiente tanto como a estudiantes, docentes, personal administrativo y usuarios externos, así contribuyendo de manera directa al proceso de llevar a lo digital procesos manuales de la ULEAM.

Recomendaciones

Al tener en cuenta las conclusiones alcanzadas en la presente investigación, se recomienda a las autoridades de la Universidad Laica Eloy Alfaro de Manabí, Matriz Manta considerar la implementación progresiva de nuestro sistema web propuesta como una herramienta institucional oficial para la gestión de escenarios universitarios. Al adoptar esto debe contemplar no solo aquel aspecto tecnológico, sino además un proceso de sensibilización y capacitación dirigido al personal administrativo y a los usuarios finales, eso con el fin de garantizar una correcta apropiación de la herramienta y maximizar sus beneficios.

Aparte se recomienda que previo a la puesta en producción del sistema, se realicen pruebas piloto controladas con un número limitado de escenarios y usuarios, permitiendo

validar la funcionalidad, usabilidad y rendimiento de la aplicación ya en un entorno real. Este tipo de proceso facilitará muchísimo la identificación temprana de posibles mejoras, ajustes técnicos o requerimientos adicionales antes de su despliegue institucional a gran escala.

Asimismo, sugerimos que nuestro sistema sea integrado con otros sistemas institucionales que se usen en la institución de manera progresiva, así como plataformas académicas, sistemas administrativos o mecanismos de autenticación corporativa. Esta interoperabilidad permitirá centralizar los datos, poder reducir la redundancia en la base de datos y mejorar demasiado la coherencia institucional, conectando con los principios de la transformación digital universitaria.

Para fases posteriores del proyecto, se aconseja tener en cuenta la inclusión de funcionalidades más sofisticadas, como reportes analíticos para facilitar la toma de decisiones, indicadores sobre el uso de los espacios, alertas automáticas y potenciales integraciones con tecnologías emergentes, por ejemplo, sensores IoT que permitirían el seguimiento en tiempo real de la ocupación de los usuarios. Con estos avances, el sistema podría transformarse en un modelo de gestión mucho más complejo y desarrollado.

Por último, recomendamos que esta tesis se use como fundamento para investigaciones académicas futuras que traten sobre la transformación digital en las universidades públicas, tener una gestión institucional y el efecto de sistemas desde una perspectiva administrativa. La experiencia que tuvimos en este proyecto puede ser replicada o adaptada a cualquier tipo de institución de educación superior del país, lo que ayudará bastante a fortalecer la calidad de gestión de estos mismos.

Anexos

Anexo 1: Cuestionario para estudiantes, docentes, personales administrativos y usuarios externos

Objetivo:

Recopilar la percepción de los estudiantes, docentes, personal administrativo y usuarios externos acerca del sistema vigente para reservar espacios en la universidad y su disposición hacía una solución digital.

Tipo de instrumento:

Cuestionario cerrado en Google forms:

- 1 = Totalmente en desacuerdo
- 2 = No estoy de desacuerdo
- 3 = Ni de acuerdo ni en desacuerdo
- 4 = Conforme
- 5 = Totalmente de acuerdo

Percepción del sistema actual.

- El proceso actual para solicitar un espacio físico en la universidad es fácil de entender.
- La información sobre los pasos para reservar un aula o auditorio es clara y accesible.
- El tiempo de respuesta para aprobar o negar una solicitud de uso de escenarios es adecuado.
- Considero que la gestión actual de los escenarios universitarios es eficiente.
- Al intentar reservar una locación universitaria, he enfrentado problemas o retrasos.
- He notado que hay conflictos de horarios o que las reservas se duplican en los escenarios universitarios.

Digitalización y uso de tecnología:

- El uso de tecnologías digitales puede mejorar los procesos administrativos de la universidad.

- La gestión manual de escenarios universitarios limita de eficiencia institucional.
- Considero necesario modernizar el proceso de reservar de escenarios mediante herramientas digitales.

Aplicación web para la gestión de escenarios:

- Me gustaría que exista una plataforma digital para hacer reservas desde mi computador.
- Un sistema web facilitaría el proceso de solicitud y aprobación de escenarios universitarios.
- Un sistema digital permitiría conocer la disponibilidad de los escenarios en tiempo real.
- Me sentiría más seguro/a utilizando un sistema digital con historial y seguimiento de solicitudes.
- Considero que una aplicación web mejoraría la transparencia en la asignación de escenarios universitarios.
- Un sistema digital reduciría errores como duplicación de reservas o pérdida de solicitudes.

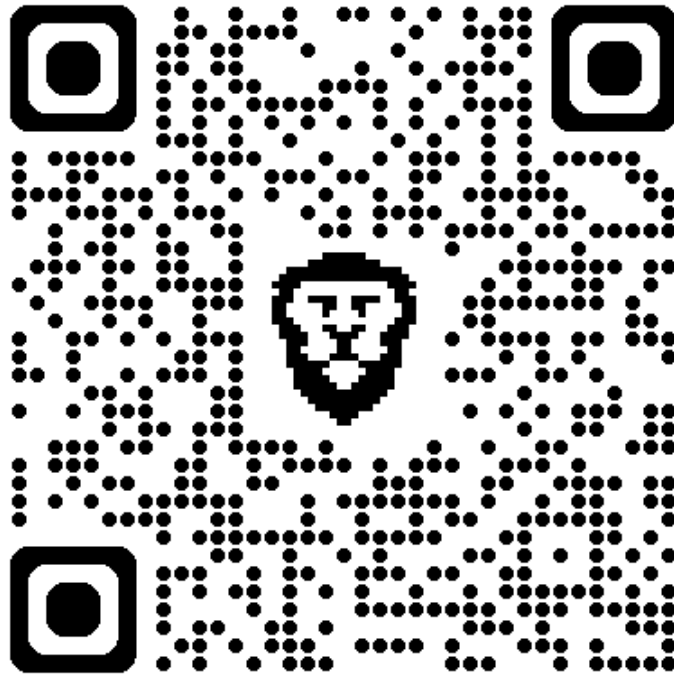
Impacto institucional y aceptación:

- La implementación de una aplicación web ayudaría a optimizar los recursos físicos de la universidad.
- Un sistema digital contribuiría a una mejor planificación de actividades académicas y administrativas.
- Recomendaría el uso de una plataforma web para reservar escenarios universitarios a otros usuarios.

Anexo 2: Enlace al instrumento de recolección de datos (Formulario Digital)

Objetivo: Facilitar el acceso al cuestionario en Google forms.

Instrucciones: Escanee el siguiente código QR para poder visualizar y responder el formulario de Google Forms utilizado en esta tesis.

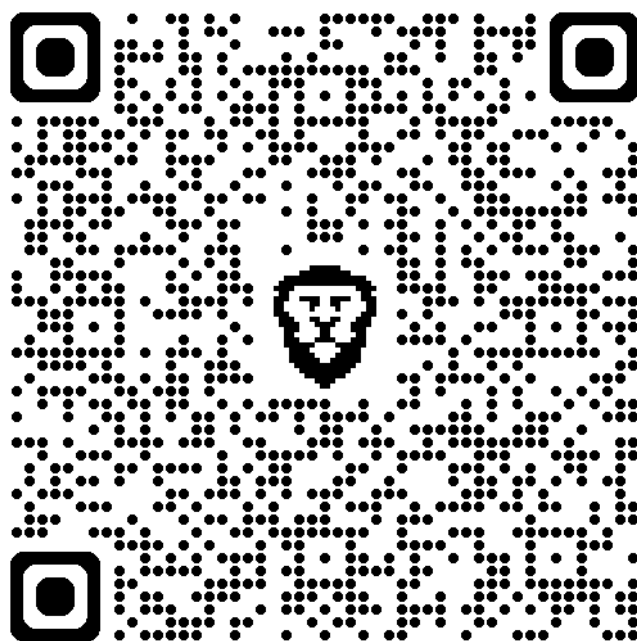


Enlace directo: <https://forms.gle/sV66vHYAbk8uJDyQ8>

Anexo 3: Acceso al Diagrama de Entidad-Relación interactivo.

Descripción: Con el fin de que los jurados y tutor puedan visualizar de mejor manera nuestro diagrama entidad-relación, se proporciona un acceso directo al modelo original mediante un código QR. Esto va a permitir que puedan explorar las relaciones, llaves y tipos de datos en alta resolución, superando las limitaciones de espacio de la versión impresa.

Instrucciones: Escanee el siguiente código QR para poder visualizar y responder el diagrama en Lucidchart.



Enlace directo: https://lucid.app/lucidchart/300d05e2-4340-43af-900c-b4773af30118/edit?viewport_loc=-1091%2C-4930%2C4411%2C2022%2C0_0&invitationId=inv_e0133125-dbc0-4ae6-98cd-9573234713b2

Anexo 4: Tutoría Guía con Docente Tutor y Personal Administrativo

Objetivo:

Adquirir datos cualitativos acerca de los procesos presentes relacionados con la gestión de escenarios, las necesidades funcionales y el grado de aceptación hacia una solución tecnológica.

Perfil de participantes:

Tutor, personal administrativo y docentes universitarios.

Preguntas guía:

¿Me podría explicar el procedimiento actual para reservar un espacio en la universidad?

¿Qué obstáculos o restricciones han detectado en dicho proceso?

¿Hay dificultades comunes como conflictos de horarios o duplicidad de reservas?

¿Cuáles cambios cree que son imprescindibles para perfeccionar la administración de escenarios?

¿Cómo sería en su perspectiva el sistema ideal?

¿Cuántos escenarios y de que tipos hay en toda la universidad?

¿Cuál es la capacidad máxima para esos espacios?

¿El sistema tiene que ser administrable?

¿Cuál sería el precio que tendríamos que cobrar a usuarios externos?

¿Cuánto es el presupuesto para hacerle mantenimiento a los usuarios?

Bibliografía

- Abrahamsson, P. (2003). Extreme Programming: First results from a controlled case study. *Proceedings of the 29th Euromicro Conference on New Waves in System Architecture (EUROMICRO'03)*, 259–266.
- Aguilar-Barojas, S. (2005). *Redalyc.org*. Retrieved from Fórmulas para el cálculo de la muestra en investigaciones de salud.: <https://www.redalyc.org/pdf/487/48711206.pdf>
- Aranda. (2013). *Aranda*. Retrieved from Aranda: <https://arandasoft.com/>
- arsys. (2025, 08 Sabado). *arsys*. Retrieved from arsys: <https://www.arsys.es/blog/php#:~:text=Conclusiones%20sobre%20PHP-,¿Qué%20es%20PHP?,web%20dinámicas%20y%20sitios%20interactivos.>
- Ávila Quiñónez, V. T. (2025). *Transformación digital en la educación superior*. Retrieved from Dominio De Las Ciencias: <https://dominiodelasciencias.com/ojs/index.php/es/article/view/4202>
- Beck, K. &. (2004, 11). *Oreilly*. Retrieved 11 22, 2025, from <https://www.oreilly.com/library/view/extreme-programming-explained/0321278658/>
- Branch, J. W. (2019). *Transformación digital en instituciones de educación superior: entre el mito y la realidad*. Bogota: Universidad Nacional de Colombia, Colombia.
- Cortés, D. (2024, Agosto 1). *blenderdeluxe*. Retrieved from blenderdeluxe: https://blenderdeluxe.com/en/laravel-tutorials/what-is-eloquent-orm-and-what-is-it-used-for-in-laravel-42?utm_source=chatgpt.com
- Doctor, A. C. (2022, 12 12). *arXiv*. Retrieved from <https://arxiv.org/abs/2212.08039>
- Doctor, A. C. (2022, 12 12). *Arxiv*. Retrieved from <https://arxiv.org/abs/2212.08039>
- Fielding, R. T. (2000, 1). *ResearchGate*. Retrieved from https://www.researchgate.net/publication/216797523_Architectural_Styles_and_the_Design_of_Network-based_Software_Architectures
- Flores-Xolocotzi, R. &. (2022, 11 23). *Decumanus*. Retrieved from <https://revistas.uacj.mx/ojs/index.php/decumanus/article/view/4914>
- Futuver. (2025). *Futuver*. Retrieved from <https://www.futuver.com/gestion-digital-clave-para-la-eficiencia-en-universidades>

- geeksforgeeks. (2024, Agosto 30). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/html/what-is-axios/>
- Janzen, D. &. (2006, 3 3). *ACM Digital Library*. doi:<https://dl.acm.org/doi/10.1145/1124706.1121419>
- Kework, M. E. (2019). *Propuesta de un sistema de control de gestión para la Universidad De La Serena Área De Pregrado*. Santiago: Univiersidad De La Serena Área De Pregrado.
- knowmad mood. (2016, Enero 26). Retrieved from knowmad mood: <https://www.knowmadmood.com/es/blog/que-es-tailwind#:~:text=En%20otras%20palabras%2C,Por%20qu%C3%A9%20utilizar%20Tailwind%20CSS?>
- Laravel. (2025, 08 Sabado). *The Powerhouse Behind Your Big Idea*. Retrieved from Laravel - The PHP Framework for Web Artisans: <https://laravel.com>
- López Velasco, A. (2024). *Diseño y desarrollo de sistema de reserva de aulas para actividades programadas utilizando Laravel y Vue.JS*. Madrid: E.T.S.I. de Sistemas Informáticos (UPM).
- Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Pearson Education.
- Medeiros, R. T. (2022). *A comparative study of web development frameworks: Angular, React and Vue.js*. Journal of Systems and Software.
- Momjian, B. (2000, 12). *PostgreSQL*. Retrieved from <https://www.postgresql.org/docs/books/>
- Morillo, G. (2018). *Sistema automatizado de gestión administrativa en el Departamento de Orientación de la Facultad de Ciencias de la Educación*. Universidad de Carabobo.
- Poimandres. (2022). *Zustand*. Retrieved from Zustand.docs: <https://zustand.docs.pmnd.rs/getting-started/introduction>
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico (7.ª ed.)*. New York: McGraw-Hill.
- Pressman, R. S. (2020). *Ingeniería de Software: Un enfoque práctico (9.ª ed.)*. McGraw-Hill Education. McGraw-Hill Education.

- Pressman, R. S. (2020). *ResearchGate*. Retrieved from https://www.researchgate.net/publication/365946272_Software_Engineering_A_Practitioner's_Approach_9_th_Edition
- R. S. Sandhu, E. J. (1996, 2). *IEEE Xplore*. doi:10.1109/2.485845
- Richards, M. (2015). *Software Architecture Patterns*. O'Reilly Media.
- Sandoval, K. Z. (2016). *Sistemas de información gerencial*. Ciudad de México: Pearson Educación.
- Sangarsu, R. R. (2019). *Journal of Scientific and Engineering Research*. Retrieved 11 22, 2025, from <https://jsaer.com/download/vol-6-iss-3-2019/JSAER2019-6-3-284-288.pdf>
- Senecyt. (2022). *Plan estrategico institucional 2021-2025*. Quito.
- Sommerville, I. (2011). *Ingeniería del Software (9.ª ed.)*. Pearson Educación.
- Stonebraker, M. &. (1986, 6 15). *ACM Digital Library*. doi:<https://dl.acm.org/doi/10.1145/16856.16888>
- Stonebraker, M. &. (1990). *The design of Postgres*. ACM SIGMOD Record.
- UNESCO. (2022). *UNESCO*. Retrieved from UNESCO: https://www.unesco.org/es/digital-education?utm_source=chatgpt.com
- UPS. (2024, 08 6). *Universidad Politécnica Salesiana*. Retrieved from Universidad Politécnica Salesiana: <https://www.ups.edu.ec/noticias?articleId=2135381&byid>
- Wohlrab, R., Pelliccione, P., Shahrokni, A., & Knauss, E. (2020). *Arxiv*. Retrieved from <https://doi.org/10.48550/arXiv.2005.09414>