

# UNIVERSIDAD TÉCNICA DEL NORTE



## Facultad De Ingeniería En Ciencias Aplicadas Carrera De Software

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

**Desarrollo de un Sistema Web para el Control de Proyectos de Construcciones en INNOVA-EP, aplicando el estándar ISO/IEC 25010 para mantenibilidad.**



**Autor:**

Ulcuango Andrimba Juan Sebastian

**Director:**

Mgs. Diego Javier Trejo España

IBARRA – ECUADOR

2026



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	1728791714		
<b>APELLIDOS Y NOMBRES:</b>	Ulcuango Andrimba Juan Sebastián		
<b>DIRECCIÓN:</b>	Cayambe, Ayora, Morona Santiago y Napo		
<b>EMAIL:</b>	<a href="mailto:jsulcuangoa@utn.edu.ec">jsulcuangoa@utn.edu.ec</a> , <a href="mailto:ulcuangoulcuango@gmail.com">ulcuangoulcuango@gmail.com</a>		
<b>TELÉFONO FIJO:</b>	SN	<b>TELÉFONO MÓVIL:</b>	0979053846

DATOS DE LA OBRA	
<b>TÍTULO:</b>	Desarrollo de un sistema web para el control de proyectos de construcción en INNOVA-EP, aplicando el estándar ISO/IEC 25010 para mantenibilidad.
<b>AUTOR (ES):</b>	Ulcuango Andrimba Juan Sebastián
<b>FECHA DE APROBACIÓN: DD/MM/AAAA</b>	31/03/2026
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
<b>TITULO POR EL QUE OPTA:</b>	INGENIERO DE SOFTWARE
<b>DIRECTOR:</b>	MSc. Trejo España Diego Javier
<b>ASESOR:</b>	MSc. Pineda Manosalvas Carpio Agapito

## CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 31 días del mes de marzo de 2026.

### **EL AUTOR:**

Firma: .....

Nombre: Juan Sebastián Ulcuango Andrimba

CI: 1728791714

## **CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

Ibarra, 31 de marzo de 2026

Mgs. Diego Javier Trejo España  
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

**CERTIFICA:**

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

---

*Mgs. Diego Javier Trejo España*

*C.C: 1002149290*

Ibarra, 27 de marzo de 2026

Ing. Cosme MacArthur Ortega Bustamante, MSc.  
**Coordinador CISIC-SW**  
**Universidad Técnica del Norte**

De mi consideración.

Por medio del presente, en mi calidad de **Gerente General** de la institución **INNOVA EP**, certifico que el estudiante **Ulcuango Andrimba Juan Sebastián**, portador de la cédula de ciudadanía N.º **1728791714**, ha culminado satisfactoriamente el proceso de desarrollo de su trabajo de integración curricular titulado **“DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE PROYECTOS DE CONSTRUCCIONES EN INNOVA-EP, APLICANDO EL ESTÁNDAR ISO/IEC 25010 PARA MANTENIBILIDAD”**.

Se informa que el sistema se encuentra operativo, funcionando con satisfacción desde el mes de enero del año 2026 constituyendo una herramienta de apoyo significativa para la modernización de nuestros procesos.

Por lo tanto, se deja constancia del cumplimiento del trabajo por parte del estudiante, para los fines académicos consiguientes ante la Universidad Técnica del Norte.

Atentamente,



MSc. Katya Lorena Bastidas Burbano  
**GERENTE GENERAL - INNOVA EP**

## **DEDICATORIA**

Dedico este trabajo a Lydia, mi madre, quien siempre me apoyó y se preocupó por mi bienestar; por demostrarme que los momentos difíciles se pueden superar y que nunca hay que rendirse en el camino.

A mi hermana Estefany, que me cuidó durante estos años de carrera, fue mi compañía y se esforzó cada día para que yo pueda culminar mis estudios universitarios, sin ella esto no sería posible.

A mi tío Marco, quien fue el referente para que yo escogiera esta carrera y que, en muchos momentos en los que necesité ayuda, siempre estuvo ahí.

A mis hermanos, que me brindaron cariño y apoyo, motivándome a ser un ejemplo para ellos, así como ellos lo han sido para mí.

A toda mi familia, porque sin cada uno de ellos, no sería la persona en la que me he convertido el día de hoy.

Ulcuango Andrimba Juan Sebastian

## **AGRADECIMIENTO**

En primer lugar, quiero agradecer a la Universidad Técnica del Norte por haberme no solo brindado una sólida formación académica y humana, si no que también me dio las mejores experiencias de mi vida.

Al ingeniero Diego Trejo, por su guía, su tiempo y por compartir sus conocimientos, los cuales fueron fundamentales para el desarrollo y culminación de este proyecto.

Agradezco también a la empresa INNOVA-EP, por abrirme las puertas, confiar en mis capacidades y brindarme las facilidades e información necesarias para implementar este sistema.

Finalmente, quiero agradecer a todos los amigos y compañeros de aula que estuvieron presentes durante estos años; en especial a Fernanda y Zamyra, porque se ganaron mi total confianza y me demostraron verdadera humildad y amistad. Muchas veces este camino fue difícil, pero ellos estuvieron ahí para hacer las cosas más llevaderas.

Ulcuango Andrimba Juan Sebastian

## ÍNDICE DE CONTENIDOS

DEDICATORIA . . . . .	VI
AGRADECIMIENTO . . . . .	VII
ÍNDICE DE FIGURAS . . . . .	XII
ÍNDICE DE TABLAS . . . . .	XIII
RESUMEN . . . . .	XIV
ABSTRACT . . . . .	XV
CAPÍTULO I	
1.1 Planteamiento del problema . . . . .	1
1.2 Objetivos . . . . .	2
1.2.1 Objetivo General . . . . .	2
1.2.2 Objetivos Específicos . . . . .	2
1.3 Alcance . . . . .	2
1.4 Metodología . . . . .	4
1.5 Justificación . . . . .	5
1.6 Antecedentes . . . . .	6
CAPÍTULO II	
2.1 Gestión de Proyectos de Construcción . . . . .	7
2.1.1 Definición y características de proyectos de construcción . . . . .	7
2.1.2 Fase del proceso constructivo y control de calidad . . . . .	8
2.1.3 Normativas y regulaciones en construcciones (Ecuador) . . . . .	10
2.2 Sistemas de Control y Supervisión en Construcción . . . . .	10

2.2.1	Sistemas tradicionales vs. Sistemas digitalizados . . . . .	11
2.2.2	Integración de datos entre organismos reguladores . . . . .	12
2.2.3	Monitoreo y seguimiento de obras civiles. . . . .	13
2.3	Desarrollo de Sistemas Web . . . . .	14
2.3.1	Arquitecturas web modernas y patrones de diseño . . . . .	15
2.3.2	Tecnologías de desarrollo: Laravel, React y PostgreSQL. . . . .	17
2.3.3	Metodologías ágiles: SCRUM para desarrollo de software . . . . .	19
2.4	Calidad de Software y Mantenibilidad . . . . .	21
2.4.1	Estándar ISO/IEC 25010: modelo de calidad de software . . . . .	21
2.4.2	Métricas de mantenibilidad . . . . .	23
2.4.3	Evaluación y medición de la calidad del software . . . . .	23
2.4.4	Modelo MANTUS como metodología de Verificación . . . . .	24
2.5	Trabajos Relacionados . . . . .	25
 CAPÍTULO III		
3.1	Definición de roles . . . . .	28
3.2	Especificación de Requisitos . . . . .	29
3.2.1	Requisitos Funcionales . . . . .	29
3.2.2	Requisitos No Funcionales . . . . .	30
3.3	Product Backlog . . . . .	31
3.3.1	Técnica de Estimación . . . . .	31
3.4	Historias de Usuario . . . . .	32
3.5	Definición de indicadores de mantenibilidad ISO/IEC 25010 . . . . .	41
3.6	Casos de Uso . . . . .	42

3.6.1	Caso 1: Gestión de proyectos . . . . .	42
3.6.2	Caso 2: Gestión de visitas . . . . .	43
3.6.3	Caso 3: Gestión de fases de construcción . . . . .	43
3.7	Arquitectura de Software . . . . .	44
3.8	Matriz de planificación . . . . .	44
3.9	Detalle de Sprints . . . . .	46
3.9.1	Sprint 1: Fundamentos y Seguridad . . . . .	46
3.9.2	Sprint 2: Gestión de Proyectos . . . . .	49
3.9.3	Sprint 3: Interoperabilidad e Integraciones . . . . .	51
3.9.4	Sprint 4: Estructuración Técnica . . . . .	54
3.9.5	Sprint 5: Flujo de Inspección . . . . .	57
3.9.6	Sprint 6: Evidencia y Ubicación . . . . .	60
3.9.7	Sprint 7: Reportes y Validación Final . . . . .	63
3.10	Resumen del Capítulo . . . . .	66
 CAPÍTULO IV		
4.1	Enfoque de Evaluación . . . . .	68
4.2	Objetivos de la Validación . . . . .	68
4.3	Verificación Técnica del Modelo MANTuS . . . . .	69
4.3.1	Cumplimiento de Prácticas de Codificación . . . . .	69
4.3.2	Check List de Subcaracterísticas . . . . .	76
4.4	Validación Experimental de la Modificabilidad . . . . .	77

4.4.1	Tareas definidas para la evaluación.....	78
4.4.2	Resultados Experimentales.....	79
4.4.3	Análisis de Tiempos de Ejecución.....	79
4.4.4	Verificación de Resultados.....	80
4.4.5	Interpretación y Discusión de Resultados.....	81
4.5	Síntesis de Resultados.....	81
4.5.1	Consolidado de Evaluaciones.....	81
4.5.2	Cumplimiento de Requisitos.....	82
	CONCLUSIONES.....	83
	RECOMENDACIONES.....	84
	Referencias	
	ANEXOS.....	92

## ÍNDICE DE FIGURAS

Figura 1	Árbol de Problemas . . . . .	1
Figura 2	Diagrama sobre el alcance del proyecto . . . . .	4
Figura 3	Diagrama de Casos de Uso: Gestión de Proyectos. . . . .	43
Figura 4	Diagrama de Casos de Uso: Gestión de Visitas. . . . .	43
Figura 5	Diagrama de Casos de Uso: Gestión de Fases de Construcción. . . . .	44
Figura 6	Arquitectura de Software del sistema INNOVA-EP. . . . .	44
Figura 7	Interfaz de inicio de sesión. . . . .	47
Figura 8	Interfaz de creación de usuarios con roles. . . . .	48
Figura 9	Interfaz de creación de proyectos. . . . .	50
Figura 10	Tabla para listar proyectos. . . . .	50
Figura 11	Vista para visualizar planos de los proyectos. . . . .	53
Figura 12	Vista para visualizar el mapa y cronogramas. . . . .	53
Figura 13	Interfaz de creación de cronogramas. . . . .	55
Figura 14	Tabla para listar cronogramas. . . . .	56
Figura 15	Interfaz de creación de inspecciones. . . . .	58
Figura 16	Calendario para mostrar visitas. . . . .	58
Figura 17	Interfaz para crear visitas desde móvil. . . . .	59
Figura 18	Interfaz para seleccionar la ubicación del proyecto. . . . .	61
Figura 19	Interfaz para seleccionar el informe de inspecciones. . . . .	62
Figura 20	Interfaz para descargar reportes de inspecciones. . . . .	64
Figura 21	Documento PDF del reporte de inspecciones. . . . .	64
Figura 22	Calendario para mostrar visitas. . . . .	65
Figura 23	Prácticas de codificación del modelo MANTUS. . . . .	67
Figura 24	Check list para cumplimiento de subcaracterísticas de mantenibilidad. . . . .	68
Figura 25	Evidencia de comentarios descriptivos en el código. . . . .	69
Figura 26	Indentación correcta en componente React. . . . .	70
Figura 27	Variables con nomenclatura descriptiva. . . . .	70
Figura 28	Estructura de archivos sin abreviaciones confusas. . . . .	71
Figura 29	División de sentencias largas en múltiples líneas. . . . .	71
Figura 30	Uso de paréntesis en lógica condicional. . . . .	72
Figura 31	Métodos con responsabilidad única. . . . .	72
Figura 32	División de lógica compleja en métodos auxiliares. . . . .	73
Figura 33	Ejemplo de función con tamaño controlado. . . . .	73
Figura 34	Control de estructuras anidadas. . . . .	74
Figura 35	Reutilización de lógica para reducir duplicidad. . . . .	74
Figura 36	Convenciones de nombres en estructura de directorios. . . . .	75
Figura 37	Funciones manejadoras de eventos (handlers) concisas. . . . .	75
Figura 38	Manejo de errores y consistencia en actualizaciones. . . . .	76

## ÍNDICE DE TABLAS

Tabla 1	Resumen de aportes relacionados al control de construcciones. . . . .	6
Tabla 2	Definición de roles del proyecto. . . . .	28
Tabla 3	Requisitos Funcionales del Sistema. . . . .	29
Tabla 4	Requisitos No Funcionales del Sistema. . . . .	30
Tabla 5	Estimación de Esfuerzo (T-Shirt Sizing). . . . .	32
Tabla 6	Historia de Usuario HU-01. . . . .	33
Tabla 7	Historia de Usuario HU-02. . . . .	34
Tabla 8	Historia de Usuario HU-03. . . . .	34
Tabla 9	Historia de Usuario HU-04. . . . .	35
Tabla 10	Historia de Usuario HU-05. . . . .	36
Tabla 11	Historia de Usuario HU-06. . . . .	36
Tabla 12	Historia de Usuario HU-07. . . . .	37
Tabla 13	Historia de Usuario HU-08. . . . .	38
Tabla 14	Historia de Usuario HU-09. . . . .	38
Tabla 15	Historia de Usuario HU-10. . . . .	39
Tabla 16	Historia de Usuario HU-11. . . . .	40
Tabla 17	Historias de Usuario Priorizadas. . . . .	40
Tabla 18	Definición de indicadores de mantenibilidad ISO/IEC 25010. . . . .	42
Tabla 19	Matriz de planificación de Sprints. . . . .	45
Tabla 20	Planificación del Sprint 1. . . . .	46
Tabla 21	Validación de Criterios de Aceptación - Sprint 1. . . . .	48
Tabla 22	Planificación del Sprint 2. . . . .	49
Tabla 23	Validación de Criterios de Aceptación - Sprint 2. . . . .	51
Tabla 24	Planificación del Sprint 3. . . . .	52
Tabla 25	Validación de Criterios de Aceptación - Sprint 3. . . . .	54
Tabla 26	Planificación del Sprint 4. . . . .	54
Tabla 27	Validación de Criterios de Aceptación - Sprint 4. . . . .	56
Tabla 28	Planificación del Sprint 5. . . . .	57
Tabla 29	Validación de Criterios de Aceptación - Sprint 5. . . . .	59
Tabla 30	Planificación del Sprint 6. . . . .	60
Tabla 31	Validación de Criterios de Aceptación - Sprint 6. . . . .	62
Tabla 32	Planificación del Sprint 7. . . . .	63
Tabla 33	Validación de Criterios de Aceptación - Sprint 7. . . . .	65
Tabla 34	Matriz de cumplimiento por subcaracterística. . . . .	76
Tabla 35	Tareas definidas para la evaluación experimental. . . . .	78
Tabla 36	Resultados consolidados de la validación experimental. . . . .	79
Tabla 37	Comparativa de Tiempos Estimados vs Reales. . . . .	80
Tabla 38	Matriz de Cumplimiento de Requisitos del Sistema. . . . .	82

## RESUMEN

La gestión eficiente de proyectos de construcción representa un pilar fundamental para el desarrollo urbano y la mitigación de riesgos asociados a la construcción informal en la provincia de Imbabura. Sin embargo, la Empresa Pública INNOVA-EP enfrentaba dificultades operativas en la planificación, ejecución y seguimiento de sus proyectos debido a la dependencia de procesos manuales, la dispersión de documentación técnica y la dificultad para registrar evidencias durante las inspecciones. Ante esta problemática, el presente trabajo de integración curricular tuvo como objetivo desarrollar un sistema web para el control de proyectos de construcciones, con criterios de ingeniería y calidad de software. Para la construcción de la plataforma se implementó una arquitectura web, utilizando el framework Laravel para la gestión de la base de datos relacional y la lógica de negocio en el Backend, y la librería React para la interfaz de usuario en el Frontend, bajo el marco de trabajo ágil Scrum. Esta solución integra funcionalidades operativas como la georreferenciación de obras, el control de paralizaciones, la gestión de convenios y la generación de reportes. Con el propósito de garantizar el funcionamiento del sistema a largo plazo y reducir la deuda técnica, el sistema fue evaluado aplicando la característica de Mantenibilidad del estándar ISO/IEC 25010, apoyado por el modelo MANTuS. El proceso de validación abarcó una evaluación estática del código y pruebas dinámicas de modificación. Los resultados arrojaron una Tasa de Éxito en el Encuentro de Fallos (TEF) del 100 % (1.0), evidenciando que el sistema posee una alta analizabilidad, modularidad y capacidad de modificación. En conclusión, la implementación de este sistema web no solo optimiza el seguimiento de las obras, sino que entrega a INNOVA-EP un producto tecnológico escalable, mantenible y preparado para evolucionar tecnológicamente.

**Palabras Clave:** Control de proyectos de construcción, Deuda técnica, ISO/IEC 25010, Laravel, Mantenibilidad, Modelo MANTuS, React, Scrum.

## ABSTRACT

The efficient management of construction projects represents a fundamental pillar for urban development and the mitigation of risks associated with informal construction in the Imbabura province. However, the Public Enterprise INNOVA-EP faced operational difficulties in the planning, execution, and monitoring of its projects due to reliance on manual processes, dispersion of technical documentation, and difficulty in recording evidence during inspections. Faced with this problem, the present curricular integration work aimed to develop a web system for the control of construction projects, applying software engineering and quality criteria. For the construction of the platform, a web architecture was implemented using the Laravel framework for relational database management and business logic in the Backend, and the React library for the user interface in the Frontend, under the Scrum agile framework. This solution integrates operational functionalities such as georeferencing of construction sites, control of work stoppages, management of agreements, and report generation. In order to guarantee the long-term operation of the system and reduce technical debt, the system was evaluated applying the Maintainability characteristic of the ISO/IEC 25010 standard, supported by the MANTuS model. The validation process included a static code evaluation and dynamic modification tests. The results yielded a Fault Encounter Success Rate (TEF) of 100 % (1.0), evidencing that the system has high analyzability, modularity, and modifiability. In conclusion, the implementation of this web system not only optimizes the monitoring of construction sites but also provides INNOVA-EP with a scalable, maintainable technological product prepared to evolve technologically.

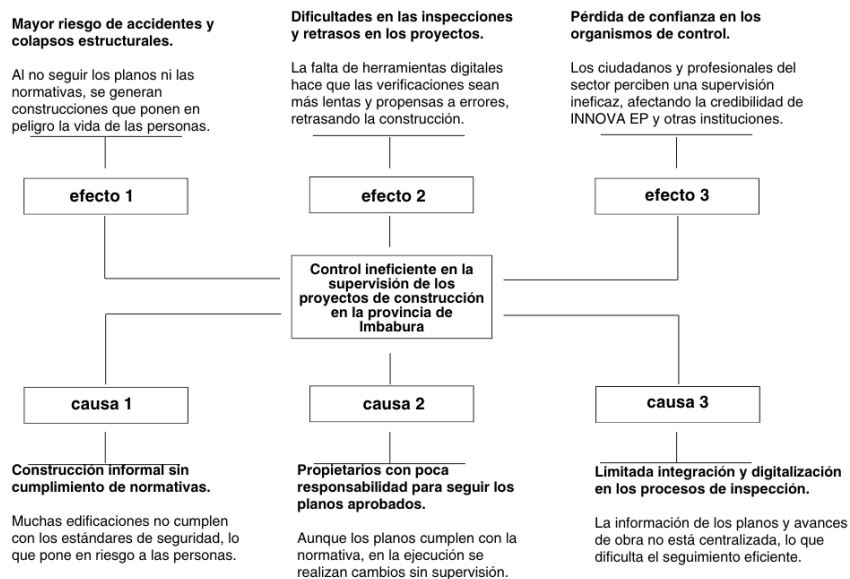
**Keywords:** Construction project control, Technical debt, ISO/IEC 25010, Laravel, Maintainability, MANTuS model, React, Scrum.

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 Planteamiento del problema

En Imbabura, la construcción informal representa un riesgo para la seguridad pública debido al control deficiente en la planificación y ejecución de edificaciones [1]. INNOVA-EP ha identificado la necesidad de implementar un sistema centralizado para registrar y validar proyectos de construcción [2]. La gestión actual enfrenta dificultades por la limitada integración entre los sistemas del COLEGIO DE INGENIEROS CIVILES DE IMBABURA, COLEGIO DE ARQUITECTOS DE ECUADOR (sede Imbabura) e INNOVA-EP [3]. Esta desconexión impide el seguimiento adecuado de las fases de construcción, mientras que los procesos manuales de validación son lentos y propensos a errores [4]. Se propone desarrollar un sistema informático que integre datos de estas tres entidades, permitiendo un control detallado de las cuatro fases de construcción [2]. Esto garantizará que las edificaciones cumplan con los planos aprobados y las normativas vigentes, mejorando la supervisión y aumentando la seguridad de las construcciones.



**Figura 1.**  
Árbol de Problemas

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

Desarrollar un Sistema Web para el Control de Proyectos de Construcciones en INNOVA-EP, aplicando el estándar ISO/IEC 25010 para mantenibilidad, con el fin de optimizar la supervisión, mejorar la validación de planos y garantizar el cumplimiento de las normativas actuales establecidas por los organismos reguladores.

### **1.2.2 Objetivos Específicos**

- Elaborar el marco teórico por medio de una revisión literaria sobre sistemas de gestión de proyectos de construcción, normas de mantenibilidad y tecnologías para desarrollo web.
- Desarrollar un sistema que permita el registro, validación y control de los proyectos de construcción usando la metodología SCRUM.
- Validar el desempeño del sistema mediante pruebas técnicas de mantenibilidad y cumplimiento de los requisitos según la norma ISO/IEC 25010.

## **1.3 Alcance**

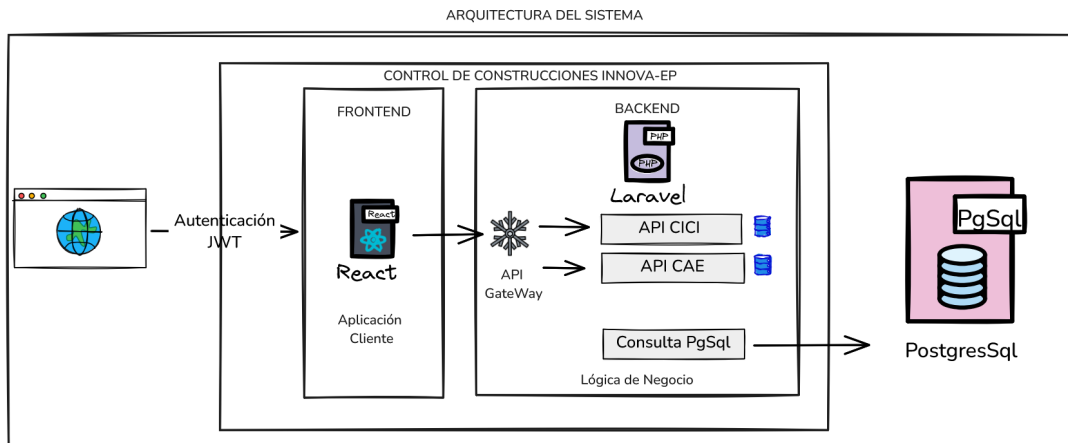
El presente trabajo de titulación tiene como propósito desarrollar un sistema web para el control de proyectos de construcción en INNOVA-EP, el cual abarcará la integración de la información proveniente del COLEGIO DE INGENIEROS CIVILES DE IMBABURA, COLEGIO DE ARQUITECTOS DE ECUADOR (sede Imbabura) y de INNOVA-EP.

El sistema permitirá la gestión de proyectos de construcción en cada una de sus cuatro fases presentes dentro del proceso del control de construcciones, mismas que recibirán un certificado que avale su cumplimiento de forma satisfactoria, desde las excavaciones hasta la supervisión final, refiriéndose al acabado del 100 % de la obra, garantizando el cumplimiento de

planos, normativas y estándares de seguridad para las edificaciones.

1. **Integración** con los sistemas de los organismos reguladores (CICI, CAE) para permitir el acceso y gestión de la información en tiempo real.
  - a. Obtener información específica y relevante de las entidades asociadas para optimizar el tiempo de respuesta de las peticiones que tendrá el sistema.
2. **Implementación** de funcionalidades para el seguimiento y control de cada fase de la construcción, asegurando que se realicen conforme a los planos aprobados.
  - a. Registrar los informes con las conclusiones estipuladas por cada inspector durante las visitas realizadas.
  - b. Definir las ubicaciones a donde se debe movilizar el equipo de inspecciones, por medio de un mapeo de los proyectos de construcción registrados en INNOVA-EP.
  - c. Gestionar las rutas de los equipos que realizarán las inspecciones para economizar el uso de recursos y tiempo.
3. **Pruebas de mantenibilidad del sistema**, garantizando la facilidad de modificación y adaptación del software para la supervisión y control de construcciones.
  - a. Realizar pruebas de modularidad para verificar que cada componente del sistema esté correctamente encapsulado, facilitando la localización y corrección de errores.
  - b. Evaluar la capacidad de integración del sistema con los datos provenientes del CICI y CAE para garantizar el acceso y actualización en tiempo real.
  - c. Validar la analizabilidad del sistema mediante pruebas que confirmen la facilidad para identificar causas de fallos o deficiencias, incluyendo la eficacia de los registros de eventos y herramientas de diagnóstico bajo distintas condiciones de carga.

Al terminar el desarrollo del sistema web, se realizará capacitaciones a los usuarios conformado por el personal de INNOVA-EP por medio de talleres y charlas, que permitirán definir el desempeño, esfuerzo, valor de precio, entre otros aspectos importantes del sistema.



**Figura 2.**

Diagrama sobre el alcance del proyecto

## 1.4 Metodología

El presente trabajo de investigación es de tipo aplicada, orientada a solucionar problemáticas en el ámbito del control de proyectos de construcción por medio de la aplicación de un software en la empresa INNOVA-EP.

El presente trabajo de investigación es de tipo aplicada, orientada a solucionar problemáticas en el ámbito del control de proyectos de construcción por medio de la aplicación de un software en la empresa INNOVA-EP.

Para el cumplimiento de los objetivos se realizarán las siguientes actividades:

1. Se analizarán documentos científicos, obtenidos de bases de datos, con el propósito de identificar marcos teóricos, metodologías aplicadas y resultados relevantes.
2. El desarrollo del sistema se ejecutará bajo el marco ágil SCRUM, organizando el trabajo en Sprints junto con la utilización de las tecnologías Laravel para la lógica del negocio, React para interfaces dinámicas y PostgreSQL por su robustez en el manejo de los datos.
3. La validación del sistema se realizará por medio de una evaluación enfocada en la mantenibilidad del producto de software, aplicando la norma ISO/IEC 25010 para evalua-

ción de las métricas de reusabilidad, modularidad, analizabilidad, capacidad de ser probado y capacidad de ser modificado [5].

## 1.5 Justificación

El desarrollo de un software de control de construcciones se justifica ante la necesidad de un seguimiento más estricto de los planos aprobados para la construcción. La solución abordará esta problemática mediante:

- **Impacto económico:** El sistema permitirá reducir costos de las obras durante el proceso de construcción, al mantener un seguimiento riguroso de los planos y cumplimiento de estos, evitando de este modo que se construyan edificaciones que no aseguran la habitabilidad de las personas.
- **Objetivo de Desarrollo Sostenible:** El proyecto se alinea con el objetivo de desarrollo sostenible número 9 Industria, Innovación e Infraestructura, y en el plan de creación de oportunidades 2021-2025 alineado a modernizar el estado [6], [7].
- **Tecnología:** Automatización de flujos de trabajo para un monitoreo y seguimiento de las obras y recursos de la empresa en tiempo real [8].

Beneficiando directamente a los empleados de la empresa INNOVA-EP, facilitándoles la realización de sus labores profesionales e indirectamente a los ciudadanos de Imbabura al apoyar a una construcción que mejora la habitabilidad de las edificaciones.

## 1.6 Antecedentes

Investigación	Aporte
<p><b>Contexto: Internacional</b> Sistema de planificación y monitorización de proyectos de obras civiles [2].</p>	<p>El artículo presenta el desarrollo de un sistema de gestión de proyectos civiles utilizando tecnologías de Microsoft como .NET 5, Blazor y Xamarin.Forms. Aporte: Desarrollar un sistema para el control de construcciones con tecnologías abiertas como Laravel, React y PostgreSQL.</p>
<p><b>Contexto: Internacional</b> Comparativa de software dedicado a la gestión de proyectos [3].</p>	<p>En el artículo analiza tres softwares de gestión de proyectos civiles identificando cuál tiene un mejor desempeño. Aporte: Desarrollo de un sistema de control de construcciones enfocado en mantenibilidad y calidad, con capacidad de evolucionar.</p>
<p><b>Contexto: Internacional</b> Implementación de un prototipo de software para el control de inicio y avance en las tareas de las obras de construcción [4].</p>	<p>En el artículo presentado se describe un prototipo de software para controlar obras de construcción, con Spring Boot y Angular. Aporte: Desarrollar un sistema de control de construcciones con Laravel, React y PostgreSQL.</p>
<p><b>Contexto: Nacional</b> Herramienta para revisión digital y captación de datos de proyectos de construcción [9].</p>	<p>En el trabajo analizado se presenta una plataforma de revisión digital de proyectos estructurales. Aporte: Desarrollo de un sistema más completo, basado en tecnologías web con facilidad de acceso y compatibilidad.</p>
<p><b>Contexto: Nacional</b> Plan de desarrollo e implementación de un sistema para el monitoreo y control de los proyectos en ejecución de la empresa S&amp;S Constructora [10].</p>	<p>En el trabajo se describe cómo los sistemas personalizados pueden mejorar la eficiencia operativa. Aporte: Desarrollo de un sistema propio de control de construcciones, aplicando la norma ISO/IEC 25010 y una solución tecnológica robusta.</p>
<p><b>Contexto: Nacional</b> Desarrollo de un software programado en MATLAB con una base de datos, para precios unitarios, presupuestos y cronogramas [11].</p>	<p>En el trabajo analizado se desarrolla un software para presupuestos y cronogramas con MATLAB y Access. Aporte: Un sistema más robusto y moderno, desarrollado en Laravel, React y PostgreSQL.</p>

**Tabla 1.**

Resumen de aportes relacionados al control de construcciones.

# CAPÍTULO II

## MARCO TEÓRICO

### 2.1 Gestión de Proyectos de Construcción

Este capítulo se centra en establecer el marco contextual, el cual va a ser la base para el desarrollo del sistema para el control de construcciones. Se explorarán los principios fundamentales de la gestión de proyectos, los sistemas de control y supervisión empleados en el sector de la construcción, las metodologías y tecnologías existentes en el desarrollo de sistemas web, y los criterios para asegurar la calidad y mantenibilidad del software resultante.

#### 2.1.1 Definición y características de proyectos de construcción

Los proyectos de construcción se constituyen en etapas complejas y multidisciplinarias que van desde la planificación pasando por el diseño y ejecución, llegando de esta forma a la entrega de las infraestructuras culminadas. De acuerdo con [12], los profesionales son los encargados de gestionar estas etapas dentro de una construcción, con el objetivo de transformar una situación actual insatisfactoria en una exitosa para el cliente. Los proyectos de construcción poseen características que están presentes a lo largo de su ejecución como:

- Un inicio y fin claramente definidos, establecidos en actividades que se deben de llevar a cabo y que deben ser supervisadas durante la ejecución del proyecto, para llegar a una situación satisfactoria para los interesados [13].
- Cada proyecto presenta condiciones particulares en términos de ubicación geográfica, diseño arquitectónico y especificaciones técnicas, lo que impide la replicación exacta entre proyectos [14].
- Se desarrollan mediante fases progresivas que incrementan gradualmente el nivel de detalle y complejidad, desde los estudios preliminares hasta los acabados finales [13].

- Requieren la coordinación eficiente entre diversos profesionales, incluyendo arquitectos, ingenieros civiles, contratistas especializados y organismos reguladores [12].

Además, los proyectos de construcción no están exentos de factores externos como características imprevistas del terreno, condiciones climáticas adversas, cambios en las normativas vigentes o aspectos internos como pueden ser errores en el diseño o ambigüedad en la información suministrada, demoras en las correcciones de los diseños e incluso falta de comunicación entre los diferentes interesados del proyecto. Esta complejidad demanda la implementación de mecanismos especializados de control y seguimiento que permitan gestionar eficientemente la cantidad de recursos que se disponen para evitar costos adicionales y garantizar la calidad del proyecto ejecutado [14], [15].

### 2.1.2 Fase del proceso constructivo y control de calidad

El proceso constructivo se estructura tradicionalmente en fases que requieren de una supervisión y un control específico, estos procesos empiezan por trabajos preliminares hasta la entrega de las edificaciones culminadas. Las fases principales del proceso constructivo incluyen:

- **Fase preliminar:** Esta etapa inicial, que incluye excavaciones y preparación del terreno, es crítica y a menudo subestimada. Los riesgos principales se asocian con estudios de suelo y ambientales deficientes, que pueden llevar a problemas de estabilidad futuros. El control de calidad se enfoca en verificar la correcta ejecución de las excavaciones según los planos topográficos, asegurar la compactación adecuada del terreno y validar que las estructuras de soporte temporal sean instaladas conforme a las normativas de seguridad [15].
- **Fase estructural:** Construcción de elementos importantes de la edificación (columnas, losas, vigas), por lo que la precisión y el cumplimiento de las normativas son vitales. Los riesgos más comunes incluyen errores en el diseño y ambigüedad en la información suministrada, así como el uso de materiales de baja calidad. La falta de una política

eficaz para la gestión de riesgos en esta etapa puede comprometer la seguridad y durabilidad de toda la infraestructura [15].

- **Fase de instalaciones y acabados:** En esta fase, la complejidad radica en la correcta coordinación de múltiples especialidades (electricidad, plomería, climatización). El control de calidad se enfoca en la inspección de las instalaciones antes de ser cubiertas por los acabados, la realización de pruebas de funcionamiento y la verificación de que los materiales de acabado cumplan con las especificaciones técnicas [15].
- **Fase de entrega:** Esta última etapa consolida todo el trabajo previo y su éxito depende de la gestión realizada durante todo el proyecto. El principal desafío es asegurar que la obra se entregue con un 100 % de avance y en total cumplimiento de las especificaciones previamente definidas y normativas [15].

La complejidad en la ejecución de proyectos de construcción ha ido aumentando a lo largo de los últimos años, lo que implica la necesidad de una mejora en el control de la calidad al momento de su realización que permita una prevención, detección y corrección efectiva de posibles deficiencias, contemplando de manera integral todas las etapas desde la planificación hasta la entrega final. Este proceso incluye la aplicación sistemática de normas técnicas, procedimientos estandarizados e inspecciones detalladas, además de la supervisión constante durante todas las fases constructivas y una redacción adecuada de los resultados obtenidos [14].

En el ámbito del control de calidad resulta fundamental designar responsables especializados que se encarguen de realizar las inspecciones técnicas en las obras. La implementación de sistemas digitales ha facilitado significativamente estos procesos, permitiendo el registro en tiempo real del estado de las inspecciones, la documentación de irregularidades identificadas y el seguimiento de las acciones correctivas implementadas [14].

### **2.1.3 Normativas y regulaciones en construcciones (Ecuador)**

El marco normativo ecuatoriano para la construcción se fundamenta principalmente en la Norma Ecuatoriana de la Construcción (NEC), establecida por el Ministerio de Desarrollo Urbano y Vivienda como el instrumento técnico que regula las actividades constructivas en el territorio nacional. Las Normas Ecuatorianas de Construcción establecen los requerimientos técnicos y metodologías que deben aplicarse al diseño sismo-resistente de edificaciones, considerando el alto potencial sísmico del Ecuador. Dentro de la NEC se encuentran normas técnicas que hacen énfasis en diseños sismo resistentes, normas para estructuras de acero, guías para el diseño estructural, mismas que tienen como propósito garantizar la habitabilidad de las obras [16].

A nivel de provincia, se evidencia dentro de Imbabura la existencia de organismos reguladores como el Colegio de Ingenieros Civiles de Imbabura (CICI) y el Colegio de Arquitectos del Ecuador sede Imbabura (CAE), que supervisan el ejercicio profesional y validan los proyectos constructivos dentro de sus respectivas competencias. Las entidades municipales como INNOVA-EP ejercen funciones de control urbano, emisión de permisos de construcción y supervisión del cumplimiento de ordenanzas locales.

Para asegurar edificaciones seguras y conformes a las regulaciones vigentes, resulta indispensable consultar la Normativa Ecuatoriana de la Construcción, que establece los estándares técnicos y de seguridad que deben cumplirse en cualquier proyecto constructivo en Ecuador [17], [18]. Esta complejidad normativa requiere sistemas de información integrados que faciliten la coordinación entre las diferentes entidades reguladoras y aseguren el cumplimiento integral de todas las normativas aplicables.

## **2.2 Sistemas de Control y Supervisión en Construcción**

Con el avance tecnológico y la evolución de las necesidades del sector, las metodologías de supervisión en los procesos constructivos han experimentado una transformación signi-

ficativa. Esta evolución se refleja en la transición de prácticas tradicionales, frecuentemente ineficientes, hacia la implementación de sistemas digitalizados que permiten una optimización considerable en la ejecución y gestión de obras [19].

### **2.2.1 Sistemas tradicionales vs. Sistemas digitalizados**

Históricamente, la gestión de proyectos de construcción ha dependido de procesos manuales tanto para la validación como para el seguimiento del estado de avance de las obras. La ausencia de herramientas digitales robustas que mejoren la eficiencia en los tiempos de verificación y validación de las obras tiene como consecuencia el incremento de los plazos de aprobación en las fases establecidas en los cronogramas de trabajo [14].

En este contexto específico de Imbabura, la presencia de construcción informal, generada por un control ineficiente en la planificación y ejecución de edificaciones, muestra la vulnerabilidad de estos enfoques manuales, afectando de forma negativa a la precisión y agilidad de los procesos operativos. Esto conlleva un riesgo para los propietarios y su seguridad al habitar edificaciones que no poseen un proceso el cual asegure su estabilidad estructural en caso de presentarse desastres naturales. Adicionalmente, la imagen de una supervisión poco efectiva por parte de ciudadanos y profesionales del sector, generada por estas malas prácticas o falta de transparencia y apertura de información, afecta a la credibilidad de las organizaciones reguladoras como INNOVA-EP, generando una desconfianza ciudadana [20].

Por otra parte, la adopción de sistemas digitales y automatizados en los flujos de trabajo en el sector de la construcción ha demostrado un potencial considerable para mejorar el seguimiento de las obras y recursos en tiempo real. Esta transición hacia sistemas digitalizados responde a la necesidad de procesos más eficientes que les brinden a los usuarios soluciones ágiles y confiables [14].

A diferencia de los sistemas tradicionales, las plataformas digitalizadas integran metodologías como BIM (Building Information Modeling), que permiten unificar la información de los proyectos en bases de datos centralizadas, facilitando un seguimiento preciso del avance de

obra a través de datos directos de la construcción [21]. Esta capacidad de unificación y análisis de datos permite minimizar las ineficiencias generadas por la dispersión de información y la dependencia de procesos manuales, promoviendo una gestión más proactiva basada en datos confiables.

### **2.2.2 Integración de datos entre organismos reguladores**

La gestión de proyectos de construcción presenta diversos desafíos, entre los cuales destaca la limitada integración de datos entre las diversas entidades reguladoras. En el contexto local de Imbabura, esta problemática se manifiesta claramente en la desconexión operativa entre el Colegio de Ingenieros Civiles de Imbabura (CICI), el Colegio de Arquitectos de Ecuador (sede Imbabura) e INNOVA-EP, en donde los procesos pasan de una entidad a otra de forma manual y en donde los propietarios se encargan de llevar la documentación requerida para que el proceso siga avanzando, generándose aún más retrasos en los plazos establecidos.

Esta falta de integración en los sistemas impide un seguimiento detallado de las fases de construcción, traduciéndose en una supervisión deficiente que afecta la confianza pública en los organismos de control [20]. Por ello, la importancia de la integración entre organismos reguladores radica en sus múltiples beneficios:

- Permite a todas las partes autorizadas acceder a la misma información, misma que estará siempre actualizada [14].
- Agiliza trámites, validaciones y aprobaciones al evitar la duplicidad de datos junto con la digitalización de estos [14].
- Proporciona datos fiables para la toma de decisiones [14].

El manejo de la información en la actualidad se realiza por parte de cada entidad, generando una complejidad innecesaria en los procesos; es por ello por lo que se opta por un manejo centralizado que permita juntar los datos de actividades que se relacionan entre sí, debido a

que hay procesos que se conectan y que pertenecen a la misma área. El uso de la tecnología permite mantener un nivel de transparencia en la información de los ciudadanos por medio de un seguimiento en tiempo real tanto de la documentación como de los procesos que están involucrados [14].

Los ciudadanos deben tener una disponibilidad de su información que les permita a su vez tener conocimiento del estado en que se encuentran los procesos que se están ejecutando. Por ello, cada entidad debe de tener una alta interoperabilidad que permita suplir la necesidad de información por parte de los ciudadanos en caso de que se presente la ausencia de una entidad y que otra pueda suplir esa información [20].

### **2.2.3 Monitoreo y seguimiento de obras civiles**

El monitoreo se enfoca en la recopilación sistemática y continua de datos sobre diversos aspectos de un proyecto durante su ejecución. El seguimiento implica el análisis de los datos recopilados para compararlos con los objetivos y metas planificadas, identificar desviaciones significativas y evaluar su impacto en el desarrollo del proyecto que muchas veces puede representar costos innecesarios y altos que afecten a la realización de los proyectos [15]. Estos procesos resultan efectivos para:

- **Detección temprana de problemas:** Permiten identificar imprevistos y posibles problemas antes de que se agraven, facilitando intervenciones oportunas que mejoren la calidad de las obras [21].
- **Optimizar el uso de recursos:** Permiten controlar el consumo de materiales, mano de obra y equipos, evitando correcciones costosas en las construcciones y aumentos en los tiempos de los cronogramas [15].
- **Mejorar la comunicación y rendición de cuentas:** Facilitan la elaboración de informes de avance precisos y la comunicación efectiva con los interesados [21].

Las técnicas y herramientas para el monitoreo y seguimiento han evolucionado significati-

vamente. Mientras que tradicionalmente se dependía de reportes manuales y comunicaciones presenciales con los interesados, actualmente se complementan y potencian con herramientas digitales avanzadas [21], [22]. El abordar temas de monitoreo de actividades con la integración de herramientas digitales permite automatizar procesos que ya han sido debidamente sistematizados, pero por su aplicación de forma tradicional, representan un coste en tiempo superior y que perjudica su eficiencia; es por ello por lo que se opta por una automatización de estos con el propósito de mejorar estas prácticas tradicionales.

El sistema propuesto facilitará el registro de informes con las conclusiones específicas de cada inspector durante las visitas realizadas, además de definir las ubicaciones hacia donde debe movilizarse el equipo de inspecciones mediante un mapeo de los proyectos. Esto constituye una digitalización integral del proceso de monitoreo de inspecciones, por medio de la integración de tecnologías modernas que contarán con funcionalidades específicas aplicadas a las etapas de planificación y control; esto provee una visión más sistemática que permite abordar inconvenientes en las fases de los procesos para ofrecer soluciones que reduzcan costos en correcciones y un cumplimiento tanto en las normativas vigentes como en los plazos establecidos [13], [22].

### **2.3 Desarrollo de Sistemas Web**

El desarrollo de un sistema web como el propuesto requiere la aplicación de arquitecturas, tecnologías y metodologías de desarrollo modernas para asegurar su funcionalidad, escalabilidad y mantenibilidad a largo plazo. La importancia de la aplicación de estas herramientas modernas nos permite obtener una buena documentación y estandarización de esta, lo cual es relevante debido a que permite un enfoque de prevención al identificar los resultados que se han obtenido y las acciones que han llevado a esos resultados [14].

### 2.3.1 Arquitecturas web modernas y patrones de diseño

Las arquitecturas web modernas se han alejado de los modelos tradicionales que se enfocaban en centralizar el trabajo y el acceso a los datos, evolucionando hacia enfoques más distribuidos y modulares. La elección de una arquitectura adecuada resulta fundamental para garantizar la flexibilidad, escalabilidad y facilidad de mantenimiento del sistema. Para que de este modo se reduzcan las tareas manuales, las cargas de trabajo, sobre costos en temas de logística aumentando la eficiencia a nivel operativo dentro de las organizaciones [19].

Entre las arquitecturas relevantes se incluyen:

- **Arquitecturas de Microservicios:** Descompone la aplicación en un conjunto de servicios pequeños y que pueden trabajar de forma autónoma e independiente. Cada servicio se centra en una capacidad de negocio específica y puede ser desarrollado y escalado sin que afecte directamente el funcionamiento de los demás. La aplicación de esta separación clara entre frontend (React) y backend (Firebase) con una API Gateway es un paso hacia el modularidad [22].
- **Aplicaciones de Página única (SPA):** Cargan una única página HTML y sin la necesidad de recargar toda la página nuevamente continúa con la carga de los elementos de acorde a como el usuario interactúa con la página principal. Esto proporciona una experiencia de usuario mucho más fluida, similar a la de las aplicaciones de escritorio o móviles. React es una tecnología fundamental para construir SPAs [22].
- **Arquitectura Orientada a API:** El diseño y desarrollo de las APIs (Interfaces de Programación de Aplicaciones) que expondrán la funcionalidad del backend se prioriza ya que son consumidas por el frontend (React) y de forma similar por otros sistemas (CICI, CAE). El uso de un API Gateway es central en este enfoque, actuando como un único punto de entrada para todas las solicitudes de los clientes y centralizando el acceso a sus datos [22].

El desarrollo de aplicaciones web modernas permite afrontar las exigencias que tienen los usuarios por medio de diseños y funcionalidades que se adaptan a las necesidades de los usuarios y que se encuentran enmarcados a través de arquitecturas y patrones de diseño. Su aplicación promueve la creación de software de mayor calidad destacando aspectos para que sea más robusto, flexible y comprensible, por medio de un enfoque metodológico y estructurado, propio de la ingeniería web destacando aspectos como la creación de sistemas interactivos con los usuarios y de alta calidad [22].

Los patrones de diseño son plantillas que describen soluciones comprobadas y documentadas las cuales permiten una optimización de tiempos en el desarrollo de software, garantizando una mayor calidad y facilitando el mantenimiento a largo plazo.

El patrón de diseño **Modelo Vista Controlador (MVC)** es un patrón que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y del módulo encargado de gestionar los eventos. Su enfoque se basa en la reutilización del código y la separación de conceptos, características que facilitan tanto el desarrollo de aplicaciones como su mantenimiento posterior [23]. Sus tres componentes son:

- **Modelo:** Es la representación de la información con la que opera el sistema. Gestiona todos los accesos a dicha información, tanto las consultas como las actualizaciones, e implementa la lógica de negocio [23].
- **Vista:** Presenta los datos del modelo en un formato adecuado para que el usuario puede interactuar con ellos, constituyendo la interfaz de usuario. Recibe los datos enviados por el controlador para luego mostrarlos a los usuarios [23].
- **Controlador:** Responde a los eventos, que usualmente son las acciones del usuario, e invoca peticiones al Modelo cuando se realiza alguna solicitud sobre la información. Actúa como un intermediario durante la comunicación entre la Vista y el Modelo [23].

### 2.3.2 Tecnologías de desarrollo: Laravel, React y PostgreSQL

La selección de tecnologías es una decisión crucial que impacta el proceso de desarrollo, el rendimiento del sistema y su mantenibilidad futura. Para este proyecto se ha elegido un conjunto de herramientas tecnológicas modernas, robustas y ampliamente usadas en la industria.

Laravel es un framework de código abierto, el cual utiliza PHP como lenguaje de programación y que dentro de su estructura de trabajo maneja la arquitectura Modelo-Vista-Controlador (MVC), se caracteriza por tener una sintaxis sencilla, facilitando el desarrollo de proyectos de forma rápida y efectiva [24]. Se seleccionó para la construcción del *backend* debido a su completo ecosistema de herramientas integradas que agilizan el desarrollo, entre las que destacan:

- **Eloquent ORM:** Es una herramienta que permite un mapeo de las relaciones en los objetos para interactuar con la base de datos de una manera orientada a objetos [24].
- **Sistema de enrutamiento:** Proporciona una gestión sencilla y clara de las URLs y las peticiones HTTP dentro del sistema [24].
- **Migraciones:** Permiten un control de versiones del esquema de la base de datos, facilitando su modificación y compartición [24].
- **Consola Artisan:** Una interfaz de línea de comandos que agiliza tareas comunes del desarrollo [24].
- La existencia de una gran comunidad de desarrolladores garantiza un soporte continuo y una amplia disponibilidad de paquetes para extender sus funcionalidades [24].

Por otra parte, React es una librería de JavaScript de código abierto, declarativa y flexible, mantenida por Facebook, que permite el desarrollo de interfaces de usuario complejas mediante pequeñas y aisladas piezas de código llamadas componentes. Su elección para el frontend se fundamenta en:

- **Alto rendimiento y escalabilidad:** React utiliza un Virtual DOM, que es una copia del DOM del navegador guardada en memoria. Este actúa como intermediario y calcula la manera más eficiente de actualizar la interfaz, renderizando solo los cambios necesarios en pantalla, lo que ahorra recursos y brinda una experiencia de usuario más fluida dentro del navegador [24].
- **Diseño modular:** La arquitectura basada en componentes reutilizables facilita el mantenimiento junto con la construcción de interfaces complejas por medio de elementos de código simples e independientes. Esto es ideal para crear las vistas dinámicas que requiere un sistema de gestión de proyectos para el personal de INNOVA-EP [24].

En un sistema que maneja grandes volúmenes de datos se hace imprescindible la utilización de una base de datos que tenga soporte para un alto nivel de concurrencia, por ello se propone la utilización de PostgreSQL, el cual es un sistema gestor de base de datos relacional orientado a objetos, de código libre y publicado bajo la licencia BSD [24]. Se eligió este motor de base de datos por sus características críticas para una aplicación que manejará datos sensibles de proyectos de construcción:

- **Integridad y robustez:** En el manejo de instrucciones es sumamente importante el cumplimiento de los estándares SQL y principios ACID, para asegurar la consistencia de los datos.
- **Gestión de cargas complejas:** Otorga un excelente control de concurrencia ya que está optimizado para manejar volúmenes de datos grandes, demostrando la escalabilidad que tendrá el sistema [24].
- **Multiplataforma y flexibilidad:** Es compatible con diversos sistemas operativos y ofrece flexibilidad para trabajar con distintos lenguajes de programación como es el caso de Laravel [24].

### 2.3.3 Metodologías ágiles: SCRUM para desarrollo de software

Las metodologías ágiles son un conjunto de enfoques que se enfocan en una construcción de proyectos de forma iterativa que priorizan la colaboración entre el equipo, una creación ágil de entregables y la flexibilidad ante los cambios. Surgen como alternativa a los modelos tradicionales, como el de cascada, que se caracterizan por concebir el proyecto con una estructura definida y un proceso secuencial rígido que no se adapta fácilmente a los cambios. El movimiento ágil se formalizó en 2001 con la publicación del "Manifiesto por el Desarrollo Ágil de Software", que establece una filosofía de trabajo basada en 4 valores y 12 principios fundamentales [25], [26].

**Scrum** es uno de los marcos de trabajo (*frameworks*) ágiles más populares y el elegido para este proyecto. Se define como un marco de trabajo que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones que se adaptan a problemas complejos. Se basa principalmente en obtener conocimiento y experiencia a partir de la práctica, junto con la aplicación de principios Lean para el uso eficiente de los recursos, centrándose en emplearlos únicamente en lo esencial. Para llevar a cabo esto, se definen periodos de trabajo que pueden ir desde un par de semanas hasta un periodo de un mes; estos ciclos de trabajo son denominados Sprints, y son aplicados con el fin de obtener un producto mínimo viable funcional que aporte valor al objetivo final del proyecto [26].

Los elementos claves de SCRUM incluyen roles que se encargan de la cohesión dentro del equipo de trabajo:

- **Product Owner:** Es la persona encargada de comunicar los intereses del cliente y de los stakeholders. Su responsabilidad principal es llevar una correcta gestión del Product Backlog, incluyendo su desarrollo, comunicación y priorización para maximizar el valor del producto [25].
- **Scrum Master:** Actúa como un líder al servicio del equipo y de la organización, asegurando que se comprendan y apliquen los principios y prácticas de Scrum; su función

principal es facilitar los eventos, eliminar obstáculos y lograr que el equipo sea completamente funcional y productivo [25].

- **Development Team:** Es un grupo de profesionales con conocimientos en áreas técnicas que construyen el incremento del producto en cada Sprint. El equipo de desarrollo es responsable de todas las actividades relacionadas con la creación del incremento, como el análisis, diseño, programación y pruebas [25].

En la metodología SCRUM los artefactos constituyen el trabajo o el valor que se genera una vez terminado el Sprint, están diseñados para maximizar la transparencia en cuanto a avances del proyecto.

- **Product Backlog:** Es un plan de trabajo representado tradicionalmente en una lista la cual muestra de forma organizada las tareas o funcionalidades para mejorar el producto [25].
- **Sprint Backlog:** Es un plan detallado creado por y para los desarrolladores. Se compone del objetivo del Sprint, el conjunto de ítems del Product Backlog seleccionados para el Sprint, y el plan para entregar el incremento [25].
- **Incremento:** Es la porción del producto completada durante un Sprint, que debe estar en condiciones de ser utilizada y aumentar el valor del producto final [25].

En el desarrollo de proyectos con SCRUM se encuentran cuatro eventos que se encargan de regular y minimizar la cantidad de reuniones no definidas que pueden existir.

- **Sprint Planning:** Es una reunión en donde todo el equipo de trabajo aporta para definir qué actividades se llevarán a cabo durante el Sprint. Se define el objetivo a completar durante la ejecución del Sprint, junto con los ítems del Product Backlog que se encuentran incluidos en el Sprint Backlog [25].

- **Daily Scrum:** Es una reunión corta de 15 minutos en donde los desarrolladores sincronizan e inspeccionan el progreso del objetivo del Sprint, además de la planificación de las siguientes 24 horas [25].
- **Sprint Review:** Se lleva a cabo una vez terminado el Sprint con el propósito de inspeccionar el resultado junto con los stakeholders para una retroalimentación [25].
- **Sprint Retrospective:** Es el último evento del Sprint y tiene como objetivo el planificar un aumento de calidad y efectividad del equipo. El equipo reflexiona sobre el sprint concluido para identificar formas de mejora a futuro [25].

El adoptar una metodología como Scrum permite que el proyecto se adapte eficientemente a los requisitos de INNOVA-EP, además de que fomenta la comunicación entre todos los involucrados aumentando la flexibilidad y productividad del equipo [25].

## **2.4 Calidad de Software y Mantenibilidad**

La calidad de un producto de software toca varias aristas, ya que no solo se limita a su funcionalidad, sino también es el grado de desempeño que tienen las características del sistema computacional durante su ciclo de vida, de modo que se garantice al cliente un sistema confiable, que satisfaga las necesidades relacionadas con la funcionalidad y eficiencia.

La mantenibilidad emerge como un factor crítico para el éxito a largo plazo del software, ya que determina la facilidad con la que alguien puede modificarlo para corregir errores, adaptarse a nuevos entornos o funcionalidades. Para una entidad como INNOVA-EP, que requiere un sistema de uso prolongado, el asegurar una alta mantenibilidad es esencial para reducir costos y facilitar su futura adaptación [5].

### **2.4.1 Estándar ISO/IEC 25010: modelo de calidad de software**

En el desarrollo de software es importante asegurar que sea construido con calidad; es por ello que se deben aplicar normativas que aseguren este factor, motivo por el cual se escoge

el estándar internacional ISO/IEC 25010. Este estándar proporciona un modelo detallado que permite un mejor enfoque en la calidad del producto al dividirlo en ocho características principales, las cuales a su vez se dividen en subcaracterísticas específicas [5], [27].

- **Adecuación funcional:** Es el nivel en que un sistema tiene las funciones que satisfacen las necesidades de acorde a las necesidades que se presentan. Incluye la completitud, corrección y pertinencia funcional [27].
- **Eficiencia de desempeño:** Es el resultado generado a partir de la cantidad de recursos y los usuarios que están solicitando una respuesta de este. Sus subcaracterísticas son el comportamiento temporal, la utilización de recursos y la capacidad [27].
- **Compatibilidad:** Se enfoca en poseer un funcionamiento estable de forma que permita un intercambio de información independientemente del entorno de hardware o software en el que se encuentre. Incluye la coexistencia y la interoperabilidad [27].
- **Usabilidad:** Es el nivel de simplicidad de un producto de software cuando es utilizado por los usuarios, enfocado en efectividad, eficiencia y satisfacción. Aquí se encuentran aspectos como el nivel de aprendizaje, la capacidad de operabilidad, protección ante errores, además de una interfaz amigable con el usuario [27].
- **Fiabilidad:** Es la capacidad para proteger la información y los datos de forma que personas o sistemas ajenos no puedan leerlos o modificarlos [27].
- **Mantenibilidad:** Efectividad y eficiencia con la que el producto puede ser modificado a futuro manteniendo una buena eficacia y eficiencia en su uso. Sus subcaracterísticas son: modularidad, reusabilidad, analizabilidad, modificabilidad y comprobabilidad [5], [28].
- **Portabilidad:** Facilidad con la que el sistema puede ser transferido de un entorno a otro. Incluye la adaptabilidad, la capacidad de instalación y la capacidad de ser reemplazado [27].

## 2.4.2 Métricas de mantenibilidad

La mantenibilidad es uno de los puntos más costosos que tiene el software, y por el mismo motivo es uno de los más importantes, ya que asegura su permanencia durante el paso del tiempo. Para evaluarla de forma objetiva, se poseen métricas asociadas a sus subcaracterísticas principales [5], [28]:

- **Modularidad:** Se enfoca en el grado en el que un sistema se fracciona en módulos de tal forma que una modificación en uno de ellos afecte de forma mínima al funcionamiento del resto.
- **Analizabilidad:** Se define como la capacidad de diagnosticar fallas o necesidades en el software y los componentes que deben ser modificados para la solución [5].
- **Modificabilidad:** Es la capacidad de generar un cambio de forma efectiva y eficiente sin que se generen defectos o afectaciones en el desempeño. Implica que el código sea fácil de entender, cambiar y probar.
- **Reusabilidad:** La capacidad de aplicar un mismo elemento como solución para un problema diferente sin mucho esfuerzo adicional; esto puede ser un componente, módulo o código y puede ser usado en más de un sistema [5].
- **Comprobabilidad:** Se refiere a qué tan factible es aplicar a un software criterios para su evaluación [5].

## 2.4.3 Evaluación y medición de la calidad del software

El evaluar la calidad de software es un proceso grande que abarca varios aspectos y que va más allá de la detección de errores. Se basa en la aplicación de un modelo de calidad que mide de forma objetiva las características del software. Para evaluar objetivamente la calidad y el cumplimiento de los estándares de codificación, no basta con la inspección manual. Por ello, el proceso general para la evaluación de la mantenibilidad de este proyecto seguirá los siguientes pasos:

- **Establecer el modelo de calidad:** Adoptar las características y subcaracterísticas de la mantenibilidad correspondiente a la ISO/IEC 25010 como base [28].
- **Definición de prácticas y métricas:** Se establecen las prácticas de codificación que contribuyen a la mantenibilidad y se seleccionan métricas para cada subcaracterística [5].
- **Elaboración de checklist y guías:** Se crea una guía con instrucciones para modificación y verificación para evaluar el cumplimiento de las subcaracterísticas [5].
- **Recolección de datos:** Se aplican las métricas al sistema para la recolección de los datos mientras se realizan las tareas de las guías, registrando tiempos y tasas de éxito [5].
- **Análisis y tabulación de resultados:** Los valores obtenidos se tabulan y comparan contra umbrales para determinar el grado de cumplimiento de los objetivos que tiene la mantenibilidad [5].
- **Informe y retroalimentación:** Se documentan los hallazgos y se proponen acciones de mejora [5].

#### 2.4.4 Modelo MANTUS como metodología de Verificación

Como proceso metodológico para la implementación y verificación de la mantenibilidad, se propone la aplicación del modelo MANTUS. Este consta de un marco referencial que aporta un conjunto de prácticas base diseñadas para integrarse al proceso de desarrollo y potenciar las subcaracterísticas de mantenibilidad de la norma ISO/IEC 25010 [29].

Se escogió el modelo MANTUS porque proporciona un proceso de validación estructurado que permite medir cuantitativamente la mantenibilidad a través de seis pasos:

- **Verificación de Prácticas Base:** Consta de una auditoría del código fuente para evidenciar y documentar el cumplimiento de las prácticas de MANTUS [5], [29].

- **Elaboración del Check List de Cumplimiento:** Elaboración de un Check List que se conecte con las prácticas implementadas y las subcaracterísticas de ISO/IEC 25010 [5], [29].
- **Elaboración de la Guía de Prueba:** Se diseña un conjunto de tareas de mantenimiento que serán entregadas a participantes externos [5], [29].
- **Recolección de Datos de los Participantes:** Se mide el desempeño de los evaluadores al realizar las tareas [5], [29].
- **Tabulación de Resultados:** Los datos obtenidos de los evaluadores se organizan y se totalizan [5], [29].
- **Aplicación de Fórmulas y Análisis:** Se calculan métricas como la tasa de éxito en el encuentro de fallos (TEF), para obtener el nivel de mantenibilidad [5], [29].

La aplicación de esta metodología permite transformar la evaluación abstracta de la mantenibilidad (basada en la norma ISO/IEC 25010) en un proceso claro que mide dicha característica y permite representar sus resultados de forma cuantitativa [5], [29].

## 2.5 Trabajos Relacionados

Gao diseñó e implementó un sistema capaz de gestionar proyectos de construcción enfocado en la web con el propósito de resolver la ineficiencia de la transmisión de información junto a una coordinación poco efectiva presente en la gestión de proyectos tradicional. El sistema fue desarrollado en una arquitectura Browser/Server (B/S), empleando JavaScript para el frontend, PHP para el backend y una base de datos MySQL. El sistema presentado fue capaz de gestionar módulos de proyectos, recursos, calidad y colaboración. Como resultado principal se obtuvo una capacidad de aproximadamente 100 usuarios de forma concurrente con un tiempo de respuesta de 1.5 segundos mediante pruebas de rendimiento. Sin embargo, el estudio presenta como limitación la ausencia de una evaluación formal de la calidad del software bajo

un estándar como la ISO/IEC 25010, centrándose en el rendimiento técnico y no en atributos de calidad como la mantenibilidad [30].

Abdurrahim desarrolló un sistema en línea para la gestión de información en el ámbito de la construcción de edificios, buscando facilitar el monitoreo remoto, la generación de informes y un manejo centralizado de la documentación de proyectos. Se aplicó la metodología cascada para el desarrollo del sistema, utilizando tecnologías como HTML, PHP y MySQL. Para la validación se aplicaron cuestionarios de satisfacción a los usuarios. El resultado más relevante fue el despliegue exitoso del sistema en línea, con una puntuación de 4.25 sobre 5. Sin embargo, una limitante encontrada fue la ausencia de pruebas de seguridad, junto con que la evaluación del sistema se basó exclusivamente en la percepción subjetiva de los usuarios, sin aplicar métricas técnicas objetivas para medir la calidad del software [31].

Nurunnisa realizó un análisis de calidad a un sistema de información de becas (BETUNAS) para evaluar y proponer mejoras a sus problemas operativos, como la lentitud y las fallas del servidor. Se siguió una metodología aplicando el estándar ISO/IEC 25010, junto con la utilización de herramientas automáticas como GTMetrix, PowerMapper y Sucuri Sitecheck para el rendimiento, compatibilidad y seguridad respectivamente. El resultado obtenido fue una calificación general de bueno, donde la usabilidad alcanzó un 78 % y la mantenibilidad fue valorada positivamente. A pesar de los resultados, el estudio se centra en realizar un diagnóstico, por lo que evaluó sin desarrollar una nueva solución o implementar las mejoras que sugiere [32].

Suparto y Dai evaluaron la calidad de un sistema de información de e-Government en Indonesia (SIRANSIJA), utilizado para la medición del desempeño laboral. La metodología se basó en la aplicación del estándar ISO/IEC 25010, midiendo sus ocho características mediante un enfoque cuantitativo con cuestionarios a 85 usuarios. El estudio concluyó que el sistema tenía un nivel de calidad bueno, obteniendo en la característica de mantenibilidad una puntuación del 73 %. Sin embargo, la principal limitación es su dependencia total de la percepción subjetiva de los usuarios, sin utilizar herramientas técnicas ni métricas objetivas del software [33].

Qazimi desarrolló un sistema web denominado OCP para la gestión digital de ofertas, contratos y pagos en empresas constructoras. Para la construcción se emplearon tecnologías como PHP, MySQL y JavaScript, y su metodología se basó en la aplicación de plantillas de texto, gestión de roles y la técnica de borrado suave (soft-delete) para asegurar la trazabilidad de los datos. El resultado fue un sistema funcional validado mediante pruebas de seguridad con herramientas como SQLMAP y Acunetix. Sin embargo, se identificó una limitación al no evaluar la calidad bajo la norma ISO/IEC 25010, enfocándose únicamente en funcionalidad y seguridad [34].

Rojas, Yaguana y Baculima desarrollaron la plataforma web SLIMFILE, orientada a la revisión digital y captación de datos de informes de diseño estructural en Ecuador. La herramienta fue desarrollada exclusivamente con tecnologías frontend (HTML, CSS, JavaScript) y fue validada de forma cualitativa mediante entrevistas a treinta ingenieros civiles. El resultado fue una plataforma funcional con valoración positiva. No obstante, carece de un backend y una base de datos para la gestión persistente, y su validación se basa únicamente en la percepción subjetiva sin una evaluación técnica formal bajo ISO/IEC 25010 [9].

Siachoque y Sierra implementaron un prototipo de software para controlar en tiempo real el avance de tareas en obras de construcción. Utilizaron una metodología de desarrollo incremental con Spring Boot para el backend y Angular para el frontend. El resultado fue un prototipo funcional con un dashboard de curvas S y estadísticas de avance. A pesar de ello, el estudio tiene como limitaciones el ser un prototipo sin validación en entorno real y la falta de un módulo de seguridad robusto y evaluación de mantenibilidad bajo estándares formales [4].

Vásquez Herrera desarrolló el software de escritorio APUFACIL.EC para automatizar la generación de análisis de precios unitarios y presupuestos. La aplicación fue programada en MATLAB con conexión a Microsoft Access. El resultado fue una herramienta funcional que optimiza el tiempo de elaboración de documentos. Sin embargo, su arquitectura de escritorio no permite el trabajo colaborativo ni es fácil de escalar, en contraste con las soluciones web modernas que permiten acceso y gestión de datos en tiempo real [11].

# CAPÍTULO III

## MATERIALES Y MÉTODOS

El presente capítulo tiene como propósito detallar las fases de desarrollo del sistema informático para INNOVA-EP. La metodología SCRUM fue utilizada para llevar a cabo este sistema, la cual aporta un marco de trabajo permitiendo una gestión ágil de tareas y tiempos, junto con el modelo de referencia MANTUS para el aseguramiento de la calidad del código.

Esta estrategia de trabajo garantiza no solo la entrega funcional de los requisitos, sino que también aporta a la incorporación de atributos de mantenibilidad según la ISO/IEC 25010, desde las etapas tempranas de programación. A continuación, se describen los roles asumidos, la planificación y la ejecución técnica de cada Sprint.

### 3.1 Definición de roles

Para el desarrollo del proyecto se definieron roles específicos que fueron asignados a los involucrados responsables de acorde a la metodología SCRUM:

**Tabla 2.**  
Definición de roles del proyecto.

<b>Nombre</b>	<b>Rol</b>	<b>Responsabilidad</b>
Mgs. Diego Trejo	Scrum Master	Encargado del seguimiento durante cada Sprint y asesoramiento.
Ing. Richard Guerrero	Product Owner	Propietario del proyecto, encargado de proporcionar los requisitos y retroalimentación.
Ulcuango Juan	Development Team	Responsable del desarrollo del proyecto y cumplimiento de los requisitos.

<b>Nombre</b>	<b>Rol</b>	<b>Responsabilidad</b>
Mgs. Carpio Pineda	Stakeholder	Interesado del proyecto encargado de validar el cumplimiento de los procesos.

### 3.2 Especificación de Requisitos

Se identificaron los requisitos funcionales y no funcionales, para garantizar la calidad en lo que debe hacer el sistema y satisfaga las necesidades dentro de la empresa.

#### 3.2.1 Requisitos Funcionales

Por medio de la metodología SCRUM, los requisitos funcionales han sido utilizados para definir las historias de usuario, en donde se detallan las necesidades desde la perspectiva del usuario final.

**Tabla 3.**  
Requisitos Funcionales del Sistema.

<b>Código</b>	<b>Nombre</b>	<b>Prioridad</b>	<b>Detalle</b>
RF1	Gestión de Convenios y Cronogramas	Alta	El sistema debe permitir registrar el convenio firmado y el cronograma de la obra presentado por el propietario, estableciendo las fechas para las futuras inspecciones.
RF2	Asignación de Recursos	Alta	El sistema debe gestionar la asignación de inspectores y vehículos disponibles para cada visita programada.
RF3	Registro de Inspecciones por Fase	Alta	El inspector debe poder registrar mediante un formulario digital los resultados de las fases del 10 %, 30 %, 70 % y 100 %, adjuntando certificados de cumplimiento.

<b>Código</b>	<b>Nombre</b>	<b>Prioridad</b>	<b>Detalle</b>
RF4	Geolocalización de Obras	Media	El sistema debe capturar las coordenadas geográficas (latitud/longitud) del proyecto para visualizar su ubicación en un mapa interactivo y planificar rutas.
RF5	Gestión de Pagos	Alta	El sistema debe calcular la cantidad a pagar según los metros cuadrados de la obra.
RF6	Control de Paralizaciones de Obras	Baja	El sistema debe permitir el registro de paralizaciones de obra y gestionar la ampliación de plazos, actualizando el cronograma si es aprobado.
RF7	Reportes Ejecutivos	Media	El sistema debe generar informes en formato PDF con el historial de inspecciones y mostrar un Dashboard con indicadores de gestión.

### 3.2.2 Requisitos No Funcionales

Los requisitos no funcionales describen los atributos de calidad, restricciones técnicas y estándares que el sistema debe cumplir.

**Tabla 4.**  
Requisitos No Funcionales del Sistema.

<b>Código</b>	<b>Nombre</b>	<b>Prioridad</b>	<b>Detalle</b>
RNF1	Accesibilidad Móvil	Alta	El sistema debe contar con una interfaz responsiva que permita su uso fluido desde dispositivos móviles para los inspectores en campo.

<b>Código</b>	<b>Nombre</b>	<b>Prioridad</b>	<b>Detalle</b>
RNF2	Seguridad de la Información	Alta	El acceso al sistema debe estar protegido mediante autenticación de usuarios y roles. La comunicación de datos sensibles debe realizarse bajo protocolo HTTPS para garantizar la confidencialidad.
RNF3	Mantenibilidad (ISO 25010)	Alta	La arquitectura del software debe cumplir con los estándares de modularidad y modificabilidad, facilitando futuras actualizaciones.
RNF4	Integridad de Datos	Alta	La base de datos debe garantizar la integridad transaccional (ACID) en el registro de inspecciones.
RNF5	Interoperabilidad	Media	El sistema debe ser capaz de consumir APIs de las entidades externas CICI y CAE para la obtención de información de los proyectos.

### **3.3 Product Backlog**

El Product Backlog está ordenado y categorizado de acorde a la prioridad de los requisitos funcionales y no funcionales del sistema. Este artefacto fue evolucionando a lo largo del proyecto para responder de forma efectiva a las necesidades de integración con las APIs externas y los estándares de control de obras.

#### **3.3.1 Técnica de Estimación**

Para determinar el esfuerzo de cada historia de usuario, se aplicó la técnica de estimación conocida como T-Shirt Sizing. Esta técnica facilita la planificación al categorizar los requisitos

en "tallas" de camisetas (XS, S, M, L, XL) basándose en la complejidad y la incertidumbre de cada tarea [35]. Gracias a esta clasificación se estableció una matriz de equivalencias en tiempo de desarrollo, agilizando la asignación de cargas de trabajo para cada Sprint.

**Tabla 5.**  
Estimación de Esfuerzo (T-Shirt Sizing).

<b>Talla</b>	<b>Complejidad</b>	<b>Estimación de Esfuerzo</b>
XS	Muy baja	1 – 4 horas
S	Baja	1 – 2 días
M	Media	2 – 3 días
L	Alta	4 – 5 días
XL	Muy Alta	1 semana

### 3.4 Historias de Usuario

Para la estimación del esfuerzo de las historias de usuario se utilizó la técnica T-Shirt Size, conocida como talla de camisetas, en el cual se especifican las siguientes categorías: XS, S, M, L, XL, correspondientes a valores de días que tomará el desarrollo de las historias de usuario. Esta herramienta es usada en marcos de trabajo ágil, con el propósito de acelerar el comienzo de los proyectos.

#### Historia de Usuario N° 1

<b>Código</b>	HU-01
<b>Nombre</b>	Gestión de Usuarios y Roles
<b>Prioridad</b>	Alta
<b>Estimación</b>	L
<b>Actor</b>	Administrador del Sistema

<b>Descripción</b>	Como administrador del sistema, quiero registrar y gestionar usuarios asignándoles roles específicos (director, recepcionista, técnico), para controlar el acceso a los módulos y a la información.
<b>Criterios de Aceptación</b>	<ol style="list-style-type: none"> <li>1. El correo electrónico debe ser único en el sistema.</li> <li>2. Al crear un usuario, debe ser obligatorio asignar un rol.</li> <li>3. Las contraseñas deben almacenarse encriptadas en la base de datos.</li> </ol>

**Tabla 6.**  
Historia de Usuario HU-01.

### Historia de Usuario N° 2

<b>ID Historia</b>	HU-02
<b>Nombre</b>	Registro y Edición de Proyectos
<b>Prioridad</b>	Alta
<b>Estimación</b>	XL
<b>Actor</b>	Recepcionista
<b>Descripción</b>	Como funcionario, quiero registrar nuevos proyectos de construcción ingresando sus datos técnicos, legales y plazos, para mantener un inventario digital de las obras bajo control.

---

**Criterios de****Aceptación**

1. El formulario debe validar que el código del proyecto sea único.
  2. Debe permitir editar la información del proyecto que no se obtenga del uso de la API.
  3. Debe validar campos obligatorios (Propietario, Ubicación).
  4. Debe permitir registrar documentos (convenios y pagos) de los proyectos.
  5. Debe permitir registrar el estado actual de las obras.
- 

**Tabla 7.**

Historia de Usuario HU-02.

**Historia de Usuario N° 3**

---

**ID Historia**

HU-03

**Nombre**

Integración con APIs Externas (CICI/CAE)

**Prioridad**

Alta

**Estimación**

M

**Actor**

Recepcionista

---

**Descripción**

Como director, quiero que el sistema muestre los documentos de planos para la verificación de las obras durante las inspecciones consultando las APIs del CICI y CAE.

---

**Criterios de****Aceptación**

1. Al seleccionar un cronograma deberá mostrar los documentos de los planos para verificar el estado de las obras durante las inspecciones.
- 

**Tabla 8.**

Historia de Usuario HU-03.

## Historia de Usuario N° 4

---

<b>ID Historia</b>	HU-04
<b>Nombre</b>	Desglose de Fases Constructivas
<b>Prioridad</b>	Media
<b>Estimación</b>	L
<b>Actor</b>	Director Técnico

---

<b>Descripción</b>	Como director, quiero visualizar el proyecto dividido en sus fases reglamentarias (Preliminar, Estructural, Acabados, Entrega), para monitorear el avance porcentual específico de cada etapa.
--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Criterios de Aceptación</b>	<ol style="list-style-type: none"><li>1. Al crear un cronograma, se deben generar automáticamente las 4 fases de control.</li><li>2. Se debe visualizar una barra de progreso independiente por cada fase.</li></ol>
--------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

**Tabla 9.**

Historia de Usuario HU-04.

## Historia de Usuario N° 5

---

<b>ID Historia</b>	HU-05
<b>Nombre</b>	Programación de Visitas
<b>Prioridad</b>	Media
<b>Estimación</b>	S
<b>Actor</b>	Director Técnico

---

<b>Descripción</b>	Como director, quiero asignar inspectores y vehículo disponibles a un proyecto específico, para una fecha determinada, organizando así la logística de control de campo.
--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

---

**Criterios de****Aceptación**

1. El sistema debe listar solo inspectores activos para asignación.
  2. El sistema debe listar solo vehículos activos para asignación.
  3. Debe permitir especificar la fecha en la que se llevara a cabo la visita.
  4. Al guardar, el proyecto debe cambiar de estado a Inspección Programada.
- 

**Tabla 10.**

Historia de Usuario HU-05.

**Historia de Usuario N° 6**

---

**ID Historia**

HU-06

**Nombre**

Registro de Resultados de Inspección

**Prioridad**

Alta

**Estimación**

XL

**Actor**

Técnico

---

**Descripción**

Como inspector, quiero registrar el informe de la visita ingresando el porcentaje de avance real y observaciones técnicas, para documentar el estado de la obra en el sistema.

---

**Criterios de****Aceptación**

1. El formulario debe ser accesible desde dispositivos móviles.
  2. Si el estado de la inspección es “con observaciones”, es obligatorio llenar el campo Observaciones.
  3. No se puede registrar una inspección con una fecha que ya pasó.
- 

**Tabla 11.**

Historia de Usuario HU-06.

## Historia de Usuario N° 7

---

<b>ID Historia</b>	HU-07
<b>Nombre</b>	Respaldo Digital de Evidencias
<b>Prioridad</b>	Media
<b>Estimación</b>	M
<b>Actor</b>	Técnico

---

<b>Descripción</b>	Como inspector, quiero subir fotografías y documentos PDF al informe de inspección, para tener un respaldo visual y legal del avance reportado.
--------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

---

<b>Criterios de Aceptación</b>	<ol style="list-style-type: none"><li>1. Solo debe permitir archivos de imagen (JPG, PNG) o documentos (PDF).</li><li>2. Debe restringir el tamaño máximo por archivo.</li><li>3. Los archivos deben poder mostrar sus nombres antes de guardar.</li></ol>
--------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

**Tabla 12.**

Historia de Usuario HU-07.

## Historia de Usuario N° 8

---

<b>ID Historia</b>	HU-08
<b>Nombre</b>	Ubicación en Mapa
<b>Prioridad</b>	Baja
<b>Estimación</b>	M
<b>Actor</b>	Técnico / Director

---

<b>Descripción</b>	Como usuario, quiero capturar la ubicación geográfica de la obra y verla en un mapa interactivo, para asegurar que la inspección se realiza en el sitio correcto y facilitar la ubicación.
--------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

---

<b>Criterios de</b>	
<b>Aceptación</b>	<ol style="list-style-type: none"> <li>1. Debe capturar latitud y longitud por medio del mapa interactivo.</li> <li>2. Debe mostrar un mapa con un marcador en la posición de la obra.</li> </ol>

---

**Tabla 13.**  
Historia de Usuario HU-08.

### Historia de Usuario N° 9

---

<b>ID Historia</b>	HU-9
<b>Nombre</b>	Generación de Ficha
<b>Prioridad</b>	Baja
<b>Estimación</b>	M
<b>Actor</b>	Director
<b>Descripción</b>	Como director, quiero descargar una ficha técnica del proyecto en formato PDF que contenga el historial de inspecciones, para entregar informes oficiales o archivar físicamente.
<b>Criterios de</b>	
<b>Aceptación</b>	<ol style="list-style-type: none"> <li>1. El PDF debe incluir el logo institucional y fecha de generación.</li> <li>2. Debe listar todas las inspecciones realizadas con sus observaciones.</li> </ol>

---

**Tabla 14.**  
Historia de Usuario HU-09.

### Historia de Usuario N° 10

---

<b>ID Historia</b>	HU-10
<b>Nombre</b>	Tablero de Control Estadístico
<b>Prioridad</b>	Baja

<b>Estimación</b>	S
<b>Actor</b>	Director
<b>Descripción</b>	Como administrador, quiero ver un panel con gráficas estadísticas sobre el estado de los proyectos (Tiempo, Retrasados, Finalizados), para tener una visión clara de los proyectos de construcción.
<b>Criterios de Aceptación</b>	<ol style="list-style-type: none"> <li>1. El Dashboard debe mostrar contadores de proyectos totales y activos.</li> <li>2. El Dashboard debe mostrar una gráfica sobre el estado de las obras.</li> <li>3. Los datos deben actualizarse al estado actual de los proyectos.</li> </ol>

**Tabla 15.**  
Historia de Usuario HU-10.

### Historia de Usuario N° 11

<b>ID Historia</b>	HU-11
<b>Nombre</b>	Verificación de Calidad
<b>Prioridad</b>	Alta
<b>Estimación</b>	L
<b>Actor</b>	Desarrollador
<b>Descripción</b>	Como equipo de desarrollo, quiero que el proceso de construcción del sistema cuente con parámetros que permitan obtener un producto basado en el modelo MANTuS y que cumple con el estándar de mantenibilidad ISO/IEC 25010.
<b>Criterios de Aceptación</b>	<ol style="list-style-type: none"> <li>1. El código debe cumplir con estándares de nomenclatura definidos.</li> </ol>

**Tabla 16.**  
Historia de Usuario HU-11.

A continuación, se estableció el listado de las historias de usuario que conformaron el alcance del sistema INNOVA-EP. Se realizó una priorización para tenerlos ordenados por su valor de negocio y dependencias técnicas.

**Tabla 17.**  
Historias de Usuario Priorizadas.

<b>Código</b>	<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Prioridad</b>	<b>Descripción</b>
HU-01	Gestión de Usuarios y Roles	L	Alta	Implementación de autenticación, middleware de seguridad y gestión de permisos.
HU-02	Registro y Edición de Proyectos	XL	Alta	Desarrollo del formulario para el ingreso de datos técnicos, legales y cronograma inicial del proyecto.
HU-03	Integración con APIs Externas (CICI/CAE)	M	Alta	Conexión HTTP con el CICI/CAE para obtención de datos de los proyectos.
HU-04	Desglose de Fases Constructivas	L	Media	Desarrollo de formularios para registro de etapas del cronograma y calcular avances.
HU-05	Programación de Visitas	S	Media	Módulo para que el director asigne uno o varios inspectores a un proyecto específico.
HU-06	Registro de Resultados de Inspección	XL	Alta	Desarrollo del formulario de visita de campo, y cálculo de avance físico real.

<b>Código</b>	<b>Historia de Usuario</b>	<b>Estimación</b>	<b>Prioridad</b>	<b>Descripción</b>
HU-07	Respaldo Digital de Evidencias	M	Media	Implementación de almacenamiento de archivos para fotos y PDFs de los informes.
HU-08	Ubicación y Visualización en Mapa	M	Baja	Integración de mapas interactivos para visualizar la ubicación exacta de los proyectos registrados.
HU-09	Generación de Ficha de Proyecto PDF	M	Baja	Generación de documentos PDF con el historial de inspecciones y estado del proyecto.
HU-10	Tablero de Control Estadístico	S	Baja	Panel gráfico para visualizar estadísticas de proyectos, cronogramas e inspecciones.
HU-11	Verificación de Mantenibilidad (Calidad)	L	Alta	Actividad transversal de refactorización, documentación técnica y ejecución de pruebas con externos.

### **3.5 Definición de indicadores de mantenibilidad ISO/IEC 25010**

Para garantizar que el sistema cumpla con el requisito no funcional de mantenibilidad, se establecieron indicadores basados en el modelo MANTUS y la norma ISO/IEC 25010. Estos indicadores permitirán una evaluación cuantitativa al finalizar cada fase de desarrollo.

**Tabla 18.**  
Definición de indicadores de mantenibilidad ISO/IEC 25010.

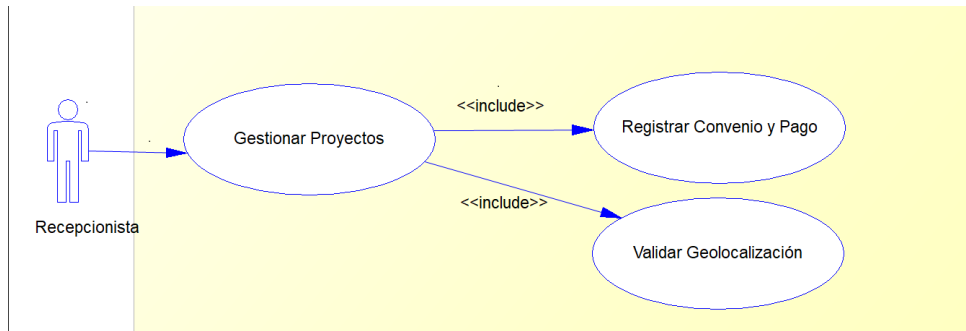
<b>Subcaracterísticas</b>	<b>Requisito</b>
Modularidad	El sistema debe estar desacoplado en capas claras (Frontend React, Backend Laravel, Base de Datos), de modo que un cambio en la interfaz no requiera modificar la lógica del servidor.
Reusabilidad	El código debe estar construido por componentes en React y Servicios en Laravel para evitar la duplicidad de lógica, permitiendo que funciones se usen en varios módulos.
Analizabilidad	El código fuente debe seguir convenciones de nombres semánticos, facilitando la lectura y comprensión rápida por parte de nuevos desarrolladores.
Capacidad de ser modificado	La implementación de nueva lógica de negocio debe localizarse en un único archivo o clase minimizando el riesgo de errores en otros archivos o funcionalidades existentes.
Capacidad de ser probado	La arquitectura debe permitir la ejecución de pruebas que permitan medir el cumplimiento de los requerimientos.

### **3.6 Casos de Uso**

Para modelar la interacción funcional entre los usuarios y el sistema, se diseñaron diagramas de Casos de Uso, que permiten visualizar el comportamiento esperado del software durante el uso de los actores principales (Administrador, Director Técnico e Inspector).

#### **3.6.1 Caso 1: Gestión de proyectos**

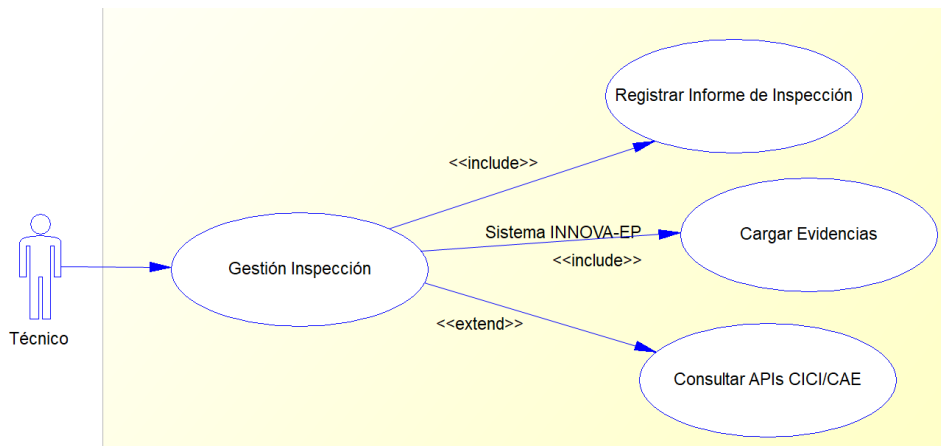
En la Figura 3 se muestran los casos de uso para la gestión de proyectos desde el usuario recepcionista.



**Figura 3.**  
Diagrama de Casos de Uso: Gestión de Proyectos.

### 3.6.2 Caso 2: Gestión de visitas

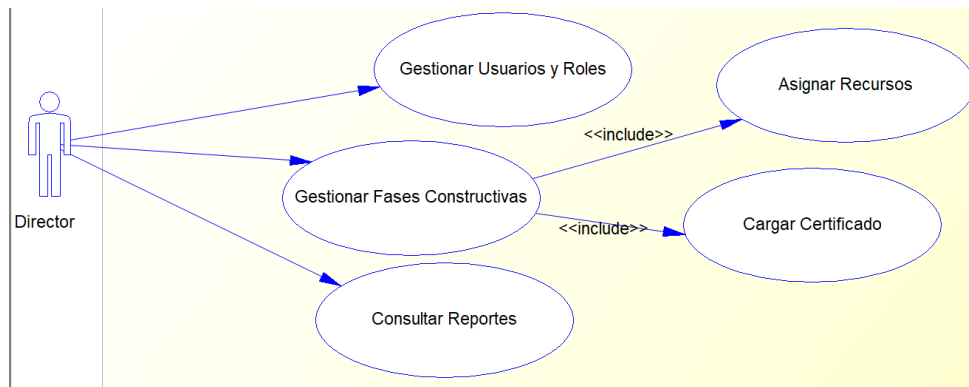
En la Figura 4 se muestran los casos de uso para la gestión de inspecciones desde el usuario técnico.



**Figura 4.**  
Diagrama de Casos de Uso: Gestión de Visitas.

### 3.6.3 Caso 3: Gestión de fases de construcción

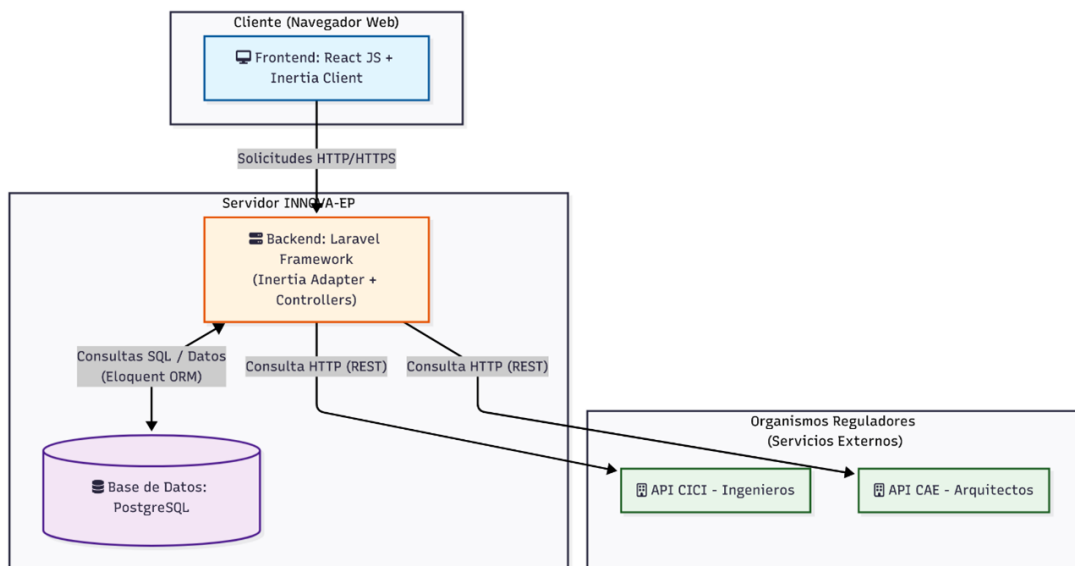
En la Figura 5 se muestran los casos de uso para la gestión de las fases de construcción, gestión de usuarios y roles y la obtención de reportes del estado de los cronogramas y visitas.



**Figura 5.**  
Diagrama de Casos de Uso: Gestión de Fases de Construcción.

### 3.7 Arquitectura de Software

El sistema INNOVA-EP se construyó con una arquitectura basada en componentes y servicios desacoplados. Como se ilustra en la Figura 6, se utiliza React JS para la capa de presentación (Frontend) comunicándose vía API REST con Laravel (Backend), el cual gestiona la lógica de negocio y la persistencia en PostgreSQL, además de manejar las conexiones con los servicios externos.



**Figura 6.**  
Arquitectura de Software del sistema INNOVA-EP.

### 3.8 Matriz de planificación

**Tabla 19.**  
Matriz de planificación de Sprints.

<b>Sprint</b>	<b>Objetivo del Sprint</b>	<b>Historias de Usuario</b>	<b>Duración</b>
Sprint 1	<b>Fundamentos y Seguridad:</b> Configuración del entorno de desarrollo, base de datos y gestión de acceso seguro (Login/Roles).	HU-01: Gestión de Usuarios y Roles.	04-Ago-25 15-Ago-25
Sprint 2	<b>Gestión de Proyectos:</b> Desarrollo del módulo central para el registro y administración de la información base de los proyectos.	HU-02: Registro de Proyectos (CRUD).	18-Ago-25 29-Ago-25
Sprint 3	<b>Interoperabilidad:</b> Conexión e integración con sistemas externos (APIs CICI y CAE) para validación de profesionales.	HU-03: Integración API CICI/CAE.	01-Sep-25 12-Sep-25
Sprint 4	<b>Estructuración Técnica:</b> Desglose detallado de los proyectos en fases constructivas para el control cronológico.	HU-05: Desglose de Fases Constructivas.	15-Sep-25 26-Sep-25
Sprint 5	<b>Flujo de Inspección:</b> Implementación de la lógica de negocio principal para la asignación de inspectores y registro de visitas.	HU-06: Programación de Visitas. HU-07: Registro de Resultados.	29-Sep-25 17-Oct-25
Sprint 6	<b>Evidencia y Ubicación:</b> Implementación de carga documental de las inspecciones y geolocalización de obras (GPS).	HU-08: Carga de Evidencia. HU-09: Ubicación en Mapa.	20-Oct-25 31-Oct-25

<b>Sprint</b>	<b>Objetivo del Sprint</b>	<b>Historias de Usuario</b>	<b>Duración</b>
Sprint 7	<b>Reportes y Validación Final:</b> Generación de salidas de información para toma de decisiones y ejecución de pruebas MANTUS.	HU-10: Ficha de Proyecto PDF.	03-Nov-25
		HU-11: Tablero de Control.	20-Nov-25
		HU-12: Verificación Calidad.	

### 3.9 Detalle de Sprints

#### 3.9.1 Sprint 1: Fundamentos y Seguridad

El primer incremento se enfocó en definir y establecer los cimientos de la arquitectura. Se configuró el entorno de desarrollo utilizando las tecnologías Laravel, React y PostgreSQL. Se implementó el módulo de seguridad que incluye autenticación, roles y permisos, junto con la definición de estándares de codificación.

#### Planificación Sprint 1

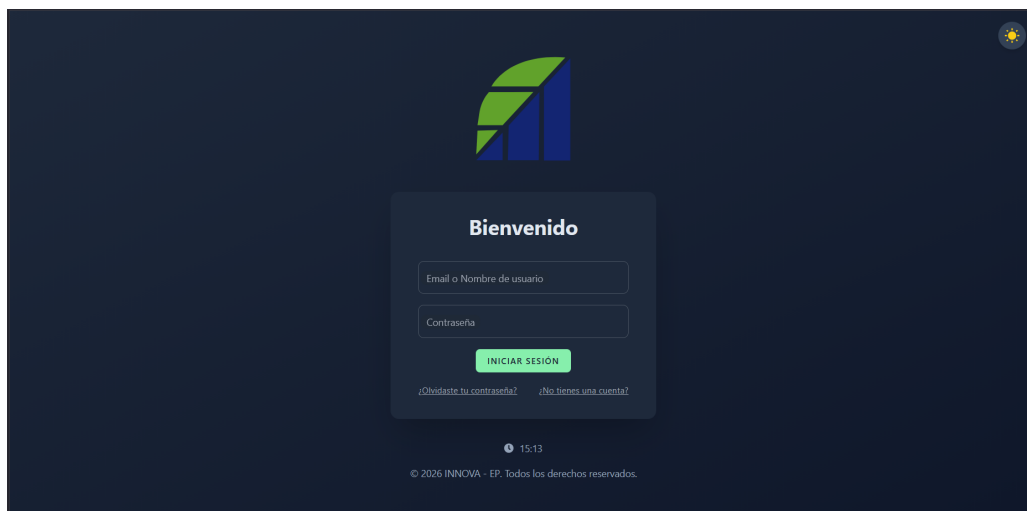
**Tabla 20.**  
Planificación del Sprint 1.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-01	Configuración	Instalación de Laravel y configuración de entorno local.	4
HU-01	Configuración	Creación del repositorio y estructura inicial en React.	2
HU-01	Base de Datos	Diseño del Modelo Entidad-Relación (Módulo Usuarios).	2
HU-01	Base de Datos	Creación de migraciones y seeders para roles y permisos.	4

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-01	Backend (API)	Desarrollo de controladores de Autenticación.	8
HU-01	Frontend	Desarrollo de Login y creación de usuarios.	8
HU-01	Validación	Pruebas de registro e inicio de sesión.	1
HU-11	MANTUS	Definición de estándares de codificación (PB-14).	1
<b>Total</b>			<b>30</b>

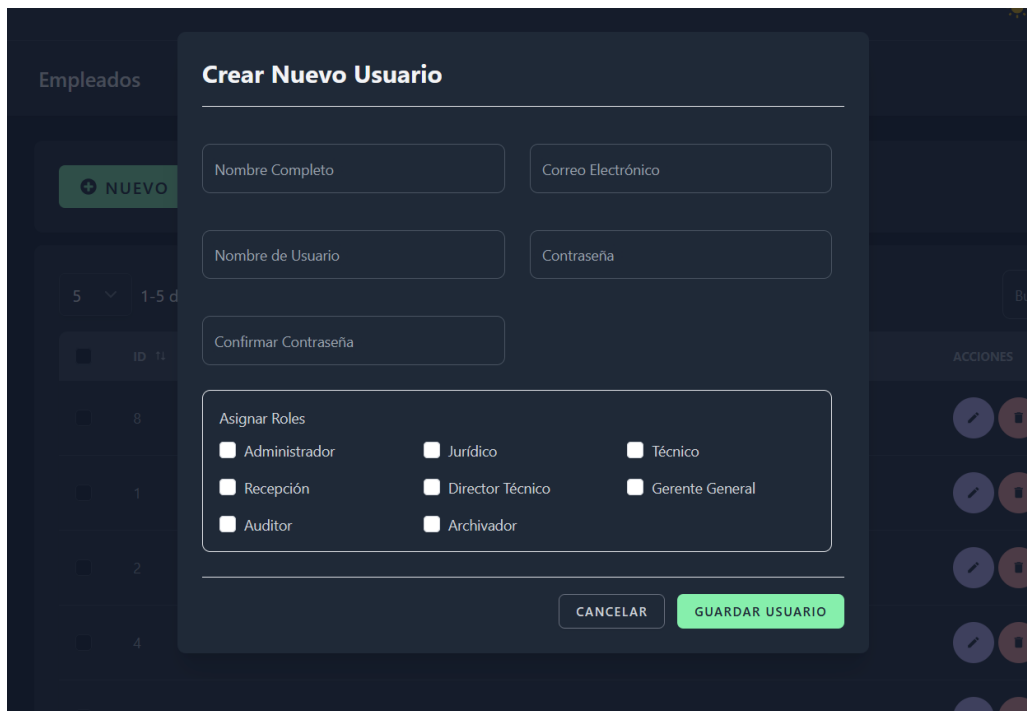
## Incremento Sprint 1

Figura 7 se observa la vista de login para inicio de sesión.



**Figura 7.**  
Interfaz de inicio de sesión.

Figura 8 se observa la vista para creación de usuarios.



**Figura 8.**  
Interfaz de creación de usuarios con roles.

## Revisión Sprint 1

**Tabla 21.**  
Validación de Criterios de Aceptación - Sprint 1.

Historia de Usuario	Criterio de Aceptación	Cumple
HU-01: Gestión de Usuarios y Roles	El correo electrónico debe ser único en el sistema.	SI
HU-01: Gestión de Usuarios y Roles	Al crear un usuario, debe ser obligatorio asignar un rol.	SI
HU-01: Gestión de Usuarios y Roles	Las contraseñas deben almacenarse encriptadas en la base de datos.	SI

## Retrospectiva Sprint 1

En la reunión retrospectiva inicial, se obtuvo una valoración positivamente al escoger separar la arquitectura en Backend (Laravel) y Frontend (React), ya que esto permitió trabajar con la lógica de negocio y la interfaz de manera independiente y ordenada. Además, se destacó la implementación de Seeders para crear los roles y permisos, esto facilitó las pruebas de autenticación.

### 3.9.2 Sprint 2: Gestión de Proyectos

Durante este Sprint de trabajo se construyó el núcleo del sistema. Se diseñó la estructura de base de datos para almacenar la información de los proyectos, se desarrollaron formularios para la captura de los datos y se aplicó la práctica de revisión de nomenclatura de variables sugerida por MANTUS para la legibilidad del código.

### Planificación Sprint 2

**Tabla 22.**  
Planificación del Sprint 2.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-02	Base de Datos	Modelado de tablas proyectos.	4
HU-02	Backend	Creación de API REST para CRUD de Proyectos.	10
HU-02	Frontend	Desarrollo de formulario de creación de proyectos (Wizard).	12
HU-02	Frontend	Implementación de tabla de listado de proyectos con filtros.	6
HU-11	MANTUS	Revisión de nomenclatura de variables (PB-05).	3
<b>Total</b>			<b>35</b>

## Incremento Sprint 2

Figura 9 se observa la vista para la creación de proyectos.

Administración de Proyectos

### AÑADIR PROYECTOS

Buscar por Folder ID

Ingrese Folder ID...

1 — 2 — 3

Datos del Proyecto (1 de 3)

Nombre del Proyecto

Dirección

Cédula del Propietario

Fecha de Registro

Estado

Metros Cuadrados

Coordenadas

Complete los campos requeridos

**Figura 9.**  
Interfaz de creación de proyectos.

Figura 10 se observa la vista para listar los proyectos registrados.

Dashboard | **Proyectos** | Cronogramas | Calendario | Mapa | Soporte

5 1-5 de 6 registros

Carpeta	Propietario	Dirección	Estado	Nombre del Proyecto	ACCIONES
2345	Diego Alejandro	Sin dirección	Ingreso	Nombre Pendiente	<input type="button" value="✎"/> <input type="button" value="✖"/>
4567	Enrique Lopez	Sin dirección	Reingreso	Nombre Pendiente	<input type="button" value="✎"/> <input type="button" value="✖"/>
9123	María José	Sin dirección	Ingreso	Nombre Pendiente	<input type="button" value="✎"/> <input type="button" value="✖"/>
5678	Anderson Ramirez	Sin dirección	Ingreso	Nombre Pendiente	<input type="button" value="✎"/> <input type="button" value="✖"/>

**Figura 10.**  
Tabla para listar proyectos.

**Tabla 23.**  
Validación de Criterios de Aceptación - Sprint 2.

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-02: Registro y Edición de Proyectos	El formulario debe validar que el código del proyecto sea único.	SI
HU-02: Registro y Edición de Proyectos	Debe permitir editar la información del proyecto que no se obtenga del uso de la API.	SI
HU-02: Registro y Edición de Proyectos	Validar campos obligatorios (Propietario, Ubicación).	SI
HU-02: Registro y Edición de Proyectos	Permitir registrar documentos (convenios y pagos) de los proyectos.	SI
HU-02: Registro y Edición de Proyectos	Permitir registrar el estado actual de las obras.	SI

## Retrospectiva Sprint 2

Durante la revisión de este Sprint, se obtuvo validaciones exitosas en los formulario de registro, ya que impiden la duplicidad de códigos de proyecto en tiempo real, mejorando la experiencia del usuario. Sin embargo, surgió un desafío relacionado con el tiempo de respuesta, ya que tardaba demasiado en validar el código del proyecto. Como mejora, se decidió ampliar el tiempo de espera para obtener la respuesta del API.

### 3.9.3 Sprint 3: Interoperabilidad e Integraciones

Este Sprint abordó uno de los desafíos técnicos más críticos del proyecto que fue la interoperabilidad. Se integró servicios para la conexión HTTP para consumir las APIs del Colegio de Ingenieros Civiles de Imbabura (CICI) y del Colegio de Arquitectos (CAE). Esto permitió obtener los datos de los proyectos, junto con la documentación de los planos.

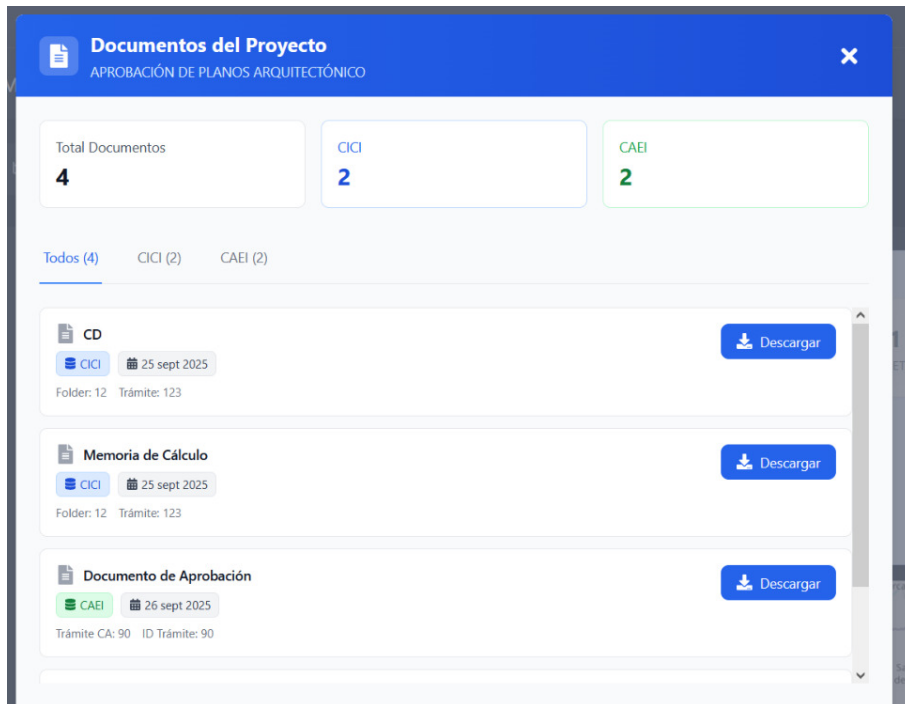
## Planificación Sprint 3

**Tabla 24.**  
Planificación del Sprint 3.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-03	Backend	Configuración de cliente HTTP en Laravel.	5
HU-03	Backend	Desarrollo de Servicios para consumo de API CICI (Ingenieros).	8
HU-03	Backend	Desarrollo de Servicios para consumo de API CAE (Arquitectos).	8
HU-03	Frontend	Implementación de autocompletado en formularios de registro.	9
HU-03	Base de Datos	Ajuste de tablas para almacenar IDs externos.	1
HU-11	MANTUS	Refactorización de servicios de integración (PB-10 Dividir métodos).	4
<b>Total</b>			<b>35</b>

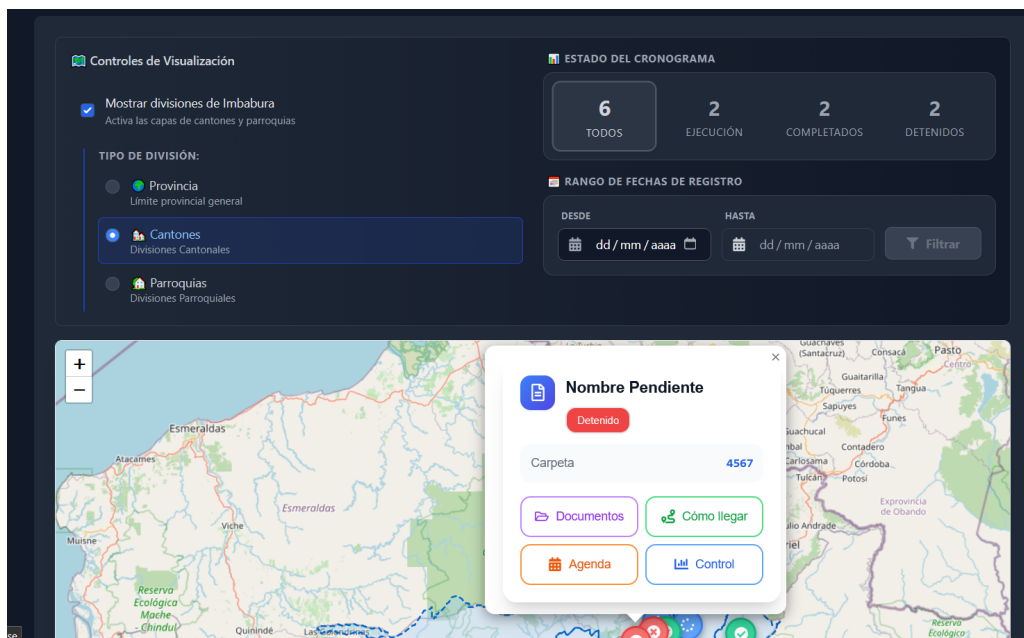
## Incremento Sprint 3

Figura 11 se observa la vista para listar los documentos de los proyectos.



**Figura 11.**  
Vista para visualizar planos de los proyectos.

Fig. 12 se observa la vista del mapa para mostrar los cronogramas.



**Figura 12.**  
Vista para visualizar el mapa y cronogramas.

**Tabla 25.**  
Validación de Criterios de Aceptación - Sprint 3.

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-03: Integración APIs CI-CI/CAE	Al seleccionar un cronograma deberá mostrar los documentos de los planos para verificar el estado de las obras durante las inspecciones.	SI

### Retrospectiva Sprint 3

Este Sprint se centró en la automatización de la carga de datos como planos y ubicaciones mediante el consumo de las APIs del CICI y CAE. Se implementó una validación para manejo de excepciones robusto con mensajes de retroalimentación en el Frontend, mejorando la percepción de rendimiento en la aplicación.

#### 3.9.4 Sprint 4: Estructuración Técnica

El objetivo de este Sprint fue implementar la lógica para el control cronológico de la obra, desglosando cada proyecto en fases constructivas estandarizadas.

### Planificación Sprint 4

**Tabla 26.**  
Planificación del Sprint 4.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-04	Base de Datos	Creación de tablas fases, items_fase y relaciones.	5
HU-04	Backend	API para gestión de fases por tipo de proyecto.	4
HU-04	Backend	Lógica para cálculo de porcentajes de avance por fase.	4

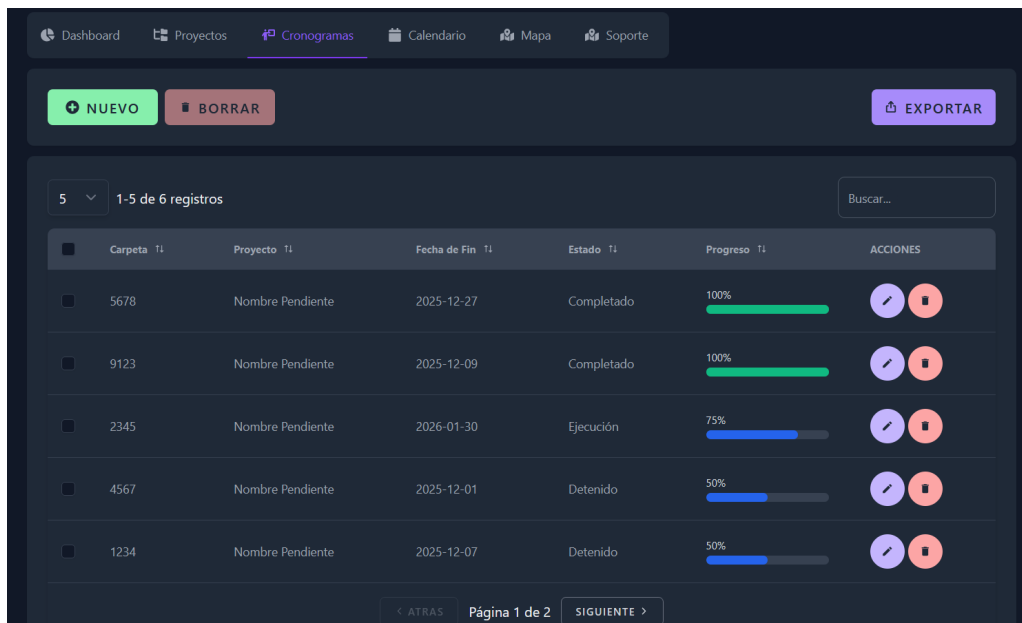
Código	Actividad	Tarea	Horas
HU-04	Frontend	Vista de detalle de proyecto con acordeón de fases.	10
HU-04	Pruebas	Verificación de cálculos de porcentajes.	2
<b>Total</b>			<b>25</b>

## Incremento Sprint 4

Fig. 14 se observa la vista para la creación de cronogramas.

**Figura 13.**  
Interfaz de creación de cronogramas.

Fig. 14 se observa la vista para listar cronogramas.



**Figura 14.**  
Tabla para listar cronogramas.

**Tabla 27.**  
Validación de Criterios de Aceptación - Sprint 4.

Historia de Usuario	Criterio de Aceptación	Cumple
HU-04: Desglose de Fases Constructivas	Al crear un cronograma, se deben generar automáticamente las 4 fases de control.	SI
HU-04: Desglose de Fases Constructivas	Se debe visualizar una barra de progreso independiente por cada fase.	SI

## Retrospectiva Sprint 4

En este Sprint se obtuvo una implementación positiva en la lógica para calcular los porcentajes de avance por fases del cronograma, eliminando posibles errores humanos en los reportes. También, se identificó que era complejo visualizar toda la estructura del cronograma en una sola pantalla, lo que saturaba la vista del usuario. Como mejora, se realizó un diseño de acordeón o pasos en la interfaz, permitiendo al usuario seleccionar solo la fase que le interesa revisar.

### 3.9.5 Sprint 5: Flujo de Inspección

Este Sprint se centró en la codificación de la lógica para el control de obras. Se desarrollaron los algoritmos para asignar inspectores, registrar avances porcentuales y determinar el cumplimiento normativo de cada fase constructiva, así como el registro de informes y evidencia del estado actual de las obras.

#### Planificación Sprint 5

**Tabla 28.**  
Planificación del Sprint 5.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-05	Base de Datos	Tablas de Asignaciones e Informes de Inspección.	6
HU-05	Backend	Controlador para asignación de inspectores.	8
HU-06	Backend	Controlador para guardar informes.	12
HU-05	Frontend	Calendario de Inspecciones para el Inspector.	12
HU-06	Frontend	Formulario para registro de inspección.	12
<b>Total</b>			<b>50</b>

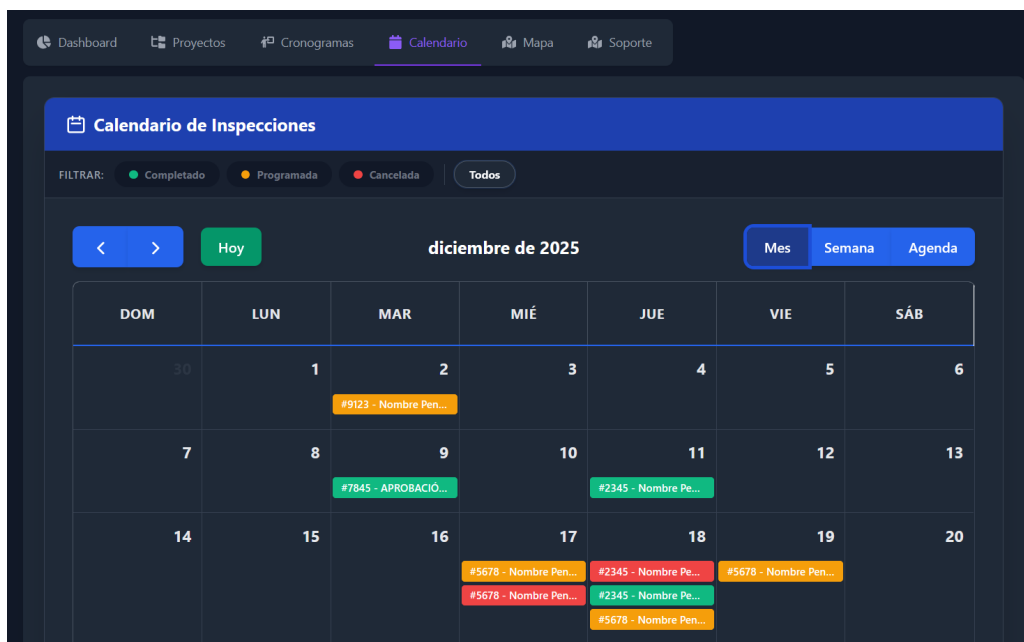
#### Incremento Sprint 5

Fig. 15 se observa la vista para crear visitas.



**Figura 15.**  
Interfaz de creación de inspecciones.

Fig. 16 se observa la vista del calendario para visitas.



**Figura 16.**  
Calendario para mostrar visitas.

Fig. 17 se observa la vista para crear visitas desde dispositivos móviles.

**Figura 17.**  
Interfaz para crear visitas desde móvil.

**Tabla 29.**  
Validación de Criterios de Aceptación - Sprint 5.

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-05: Programación de Visitas	El sistema debe listar solo inspectores activos para asignación.	SI
HU-05: Programación de Visitas	El sistema debe listar solo vehículos activos para asignación.	SI
HU-05: Programación de Visitas	Al guardar la asignación, el proyecto debe cambiar de estado a “Inspección Programada”.	SI
HU-05: Programación de Visitas	Debe permitir especificar la fecha en la que se llevara a cabo la visita.	SI
HU-06: Registro de Resultados	El formulario de inspección debe ser accesible y responsive desde móviles.	SI

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-06: Registro de Resultados	Si el estado es “con observaciones”, es obligatorio llenar el campo Observaciones.	SI
HU-06: Registro de Resultados	No se puede registrar una inspección con una fecha que ya pasó.	SI

## Retrospectiva Sprint 5

En este Sprint se evidenció un correcto diseño Responsive, dado que los inspectores validaron que la interfaz es fácil para ser usada en teléfonos durante el trabajo de campo. Un inconveniente detectado fue la gestión de las fechas en el calendario y asignación de fechas pasadas, para corregir esto, se ajustaron las reglas de validación en el Backend y se bloqueó visualmente la selección de fechas no hábiles en el componente de calendario, asegurando una correcta planificación.

### 3.9.6 Sprint 6: Evidencia y Ubicación

Para mejorar el control de obras, se integraron servicios de mapas y almacenamiento de archivos.

## Planificación Sprint 6

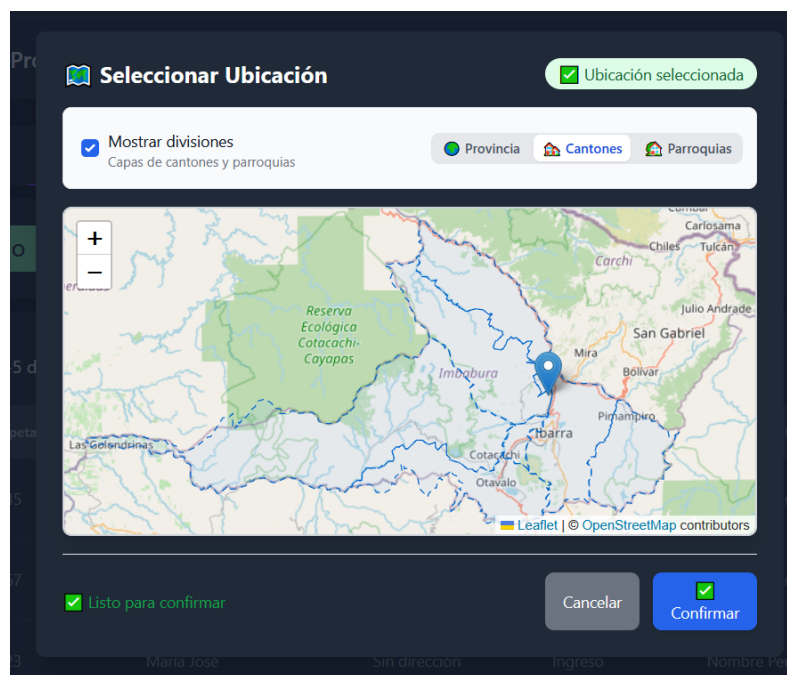
**Tabla 30.**  
Planificación del Sprint 6.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-08	Frontend	Integración de librería de mapas (Leaflet).	10
HU-07	Backend	Configuración de almacenamiento para archivos.	6

Código	Actividad	Tarea	Horas
HU-07	Frontend	Componente para carga de PDFs.	5
HU-08	Base de Datos	Campos latitud, longitud y tabla archivos_adjuntos.	8
HU-08	Pruebas	Verificación de visualización correcta de marcadores en mapa.	1
<b>Total</b>			<b>30</b>

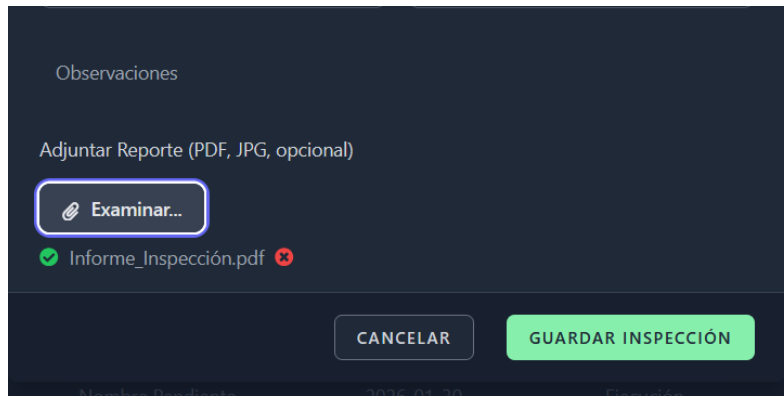
## Incremento Sprint 6

Fig. 18 se observa la vista para crear inspecciones desde dispositivos móviles.



**Figura 18.**  
Interfaz para seleccionar la ubicación del proyecto.

Fig. 19 se observa la vista para agregar informes de las inspecciones.



**Figura 19.**

Interfaz para seleccionar el informe de inspecciones.

**Tabla 31.**

Validación de Criterios de Aceptación - Sprint 6.

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-07: Respaldo Digital de Evidencias	Solo debe permitir subir archivos de imagen (JPG, PNG) y documentos (PDF).	SI
HU-07: Respaldo Digital de Evidencias	Debe restringir el tamaño máximo por archivo.	SI
HU-07: Respaldo Digital de Evidencias	El archivo debe poder mostrar su nombre antes de guardar.	SI
HU-08: Ubicación en Mapa	Debe capturar latitud y longitud por medio del mapa interactivo.	SI
HU-08: Ubicación en Mapa	Debe mostrar un mapa con un marcador en la posición exacta de la obra.	SI

## Retrospectiva Sprint 6

Durante este Sprint se destacó la integración de los mapas interactivos como una de las funcionalidades de mayor valor visual. También, se integro segmentaciones para evidenciar a la provincia de Imbabura, sus cantones y parroquias. Para lograr esto, se implementó los datos de georeferencia y controles en el frontend para gestionar la segmentación del mapa.

### 3.9.7 Sprint 7: Reportes y Validación Final

La etapa final se centró en la explotación de datos para la toma de decisiones y la preparación para la validación externa.

#### Planificación Sprint 7

**Tabla 32.**  
Planificación del Sprint 7.

<b>Código</b>	<b>Actividad</b>	<b>Tarea</b>	<b>Horas</b>
HU-09	Backend	Implementación de generador PDF.	10
HU-10	Frontend	Diseño de Dashboard con gráficas estadísticas.	8
HU-11	MANTUS	Preparación de Guía de Modificación y Entorno de Pruebas.	8
HU-11	MANTUS	Ejecución de pruebas con participantes externos.	12
HU-11	MANTUS	Tabulación de resultados y cálculo de TEF.	7
<b>Total</b>			<b>45</b>

#### Incremento Sprint 7

Fig. 20 se observa la vista para descargar reportes de las inspecciones.



**Figura 20.**  
Interfaz para descargar reportes de inspecciones.

Fig. 21 se observa el informe PDF de las inspecciones.

### Control del Cronograma #2345

Fecha: 11/1/2026  
Reporte realizado por: Usuario Director



#### Información del Control

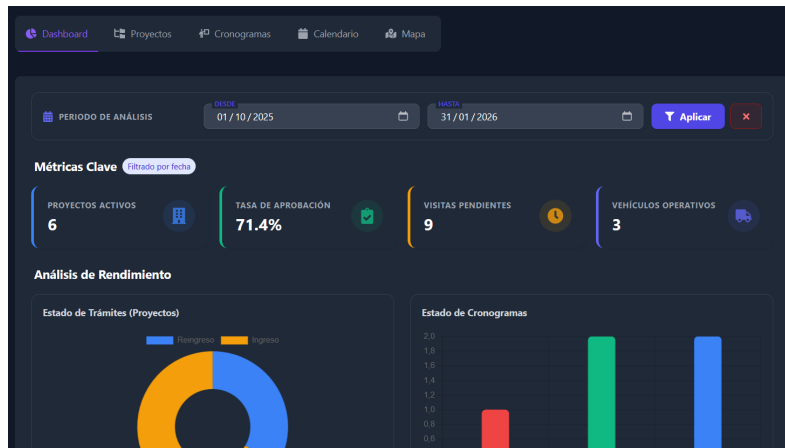
**Número de Control:** Control #1  
**Descripción:** Control del 10% de la obra  
**Fecha Planificada:** 2025-11-28  
**Estado:** Completado

#### Inspecciones Registradas (6)

Fecha	Resultado	Observaciones	Inspectores	Vehiculos
2026-01-10	Aprobado	Cumple con lo acordado en los planos	Usuario Técnico	0002 - Volsvagen
2026-01-09	Rechazado	Retraso de acorde a los cronogramas	Usuario Director	159 - Corola
2026-01-07	Sin resultados	Pendiente	Usuario Técnico	0002 - Volsvagen

**Figura 21.**  
Documento PDF del reporte de inspecciones.

Fig. 22 se observa la vista de gráficas estadísticas sobre los proyectos.



**Figura 22.**  
Calendario para mostrar visitas.

**Tabla 33.**  
Validación de Criterios de Aceptación - Sprint 7.

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumple</b>
HU-09: Generación de Ficha PDF	El PDF debe incluir el logo institucional y fecha de generación.	SI
HU-09: Generación de Ficha PDF	Debe listar todas las inspecciones realizadas con sus observaciones.	SI
HU-10: Tablero de Control	El Dashboard debe mostrar contadores de proyectos totales y activos.	SI
HU-10: Tablero de Control	El Dashboard debe mostrar una gráfica sobre el estado de las obras.	SI
HU-10: Tablero de Control	Los datos deben actualizarse al estado actual de los proyectos.	SI
HU-11: Verificación de Calidad	El código debe cumplir con los estándares de nomenclatura definidos (MANTUS).	SI

## **Retrospectiva Sprint 7**

En la reunión de cierre del desarrollo, se concluyó con éxito la generación de reportes PDF, con datos de los controles y las inspecciones realizadas. Se ajustaron los diseños de los reportes para la generación de documentos, se logró una plantilla con un diseño que abarca toda la información de los controles e inspecciones. Finalmente, se resalto la aplicación de las prácticas del modelo MANTUS durante todos los Sprints anteriores lo cual permitió obtener un resultado con calidad y mantenibilidad para el producto final.

### **3.10 Resumen del Capítulo**

La ejecución de las siete iteraciones planificadas permitió transformar los requisitos iniciales en un producto software funcional y robusto. La aplicación de la metodología SCRUM facilitó la adaptación ante los desafíos de integración con APIs externas, mientras que la incorporación de las prácticas del modelo MANTUS como la modularidad y los estándares de codificación, aseguró que el código fuente realizado posea un alto nivel de mantenibilidad. El sistema para INNOVA-EP se encuentra, listo para la fase de validación dinámica y estática, cuyos resultados se detallan en el siguiente capítulo.

# CAPÍTULO IV

## RESULTADOS Y ANÁLISIS

El presente capítulo tiene como objetivo mostrar los resultados obtenidos una vez culminado el desarrollo del sistema para el control de construcciones para INNOVA-EP. Con la finalidad de validar el cumplimiento de la ISO/IEC 25010 y sus características enfocadas en la Mantenibilidad se aplicó el modelo MANTuS, que proporciona una guía de aspectos que debe de cumplir el proceso de construcción, mismos aspectos que generan calidad, reducen costos y ayudan a la eficiencia en el equipo de desarrollo.

Prácticas base
DEV-MANT.4.PB1: Utilizar comentarios. Hacer uso de comentarios adecuados en el código fuente. [Resultados: a, c, d]
DEV-MANT.4.PB2: Describir propósitos. Explicar el propósito de las funciones, subrutinas, variables y constantes. [Resultados: a, c, d]
DEV-MANT.4.PB3: Verificar los flujos de control. Incluir solamente los flujos de control necesarios. [Resultados: b, c, e, f]
DEV-MANT.4.PB4: Organizar código. El código fuente debe ser indentado. [Resultado: c, d]
DEV-MANT.4.PB5: Asignar nombres claros. Dar nombres descriptivos a las variables. [Resultado: c, d]
DEV-MANT.4.PB6: Controlar abreviaciones. Hacer poco uso de abreviaciones. [Resultado: c, d]
DEV-MANT.4.PB7: Controlar sentencias. Evitar sentencias largas en el código fuente, es mejor mantener las líneas de código cortas, aunque esto implica separar las sentencias en múltiples líneas. [Resultado: c, d]
DEV-MANT.4.PB8: Controlar paréntesis. Hacer correcto uso de los paréntesis para mejorar la facilidad de lectura de las expresiones aritméticas y lógicas. [Resultado: c, d]
DEV-MANT.4.PB9: Determinar funcionalidades. Implementar solamente las funcionalidades necesarias. [Resultados: b, c, e, f]
DEV-MANT.4.PB10: Dividir métodos. Mantener métodos simples, es decir, dividir los métodos que tengan muchas condiciones. [Resultados: b, c, e, f]
DEV-MANT.4.PB11: Controlar tamaño. Evitar que el tamaño del código crezca innecesariamente. [Resultados: c, f]
DEV-MANT.4.PB12: Controlar nivel de anidación. Evitar las estructuras de control anidadas innecesarias, ya que estas son más complejas que las estructuras de control secuenciales. [Resultados: b, c, e, f, g]
DEV-MANT.4.PB13: Evitar duplicación. Detectar código duplicado, extrayéndolo en un nuevo procedimiento y reemplazando todas las instancias del código duplicado por llamadas al nuevo procedimiento. [Resultados: b, e, f, i]
DEV-MANT.4.PB14: Definir estándares. Tener un conjunto de estándares de programación en la escritura de código para evitar la individualidad entre los programadores. [Resultados: d, h]
DEV-MANT.4.PB15: Utilizar convenciones. Hacer uso de convenciones estándar para el nombrado de paquetes, clases, métodos y variables. [Resultados: d, h]
DEV-MANT.4.PB16: Controlar tamaño de unidad. Las piezas de funcionalidad de más bajo nivel deben mantenerse pequeñas para que sean centradas y fáciles de entender. [Resultados: b, c, e, f]
DEV-MANT.4.PB17: Verificar trazabilidad. Verificar periódicamente que los artefactos de una etapa sean consistentes con artefactos de la etapa anterior. [Resultados: c, j, k]
DEV-MANT.4.PB18: Actualizar artefactos. Actualizar los artefactos cuando se presenten cambios. [Resultados: c, j, k]
Producto de trabajo de salida
PTS-06 Unidad de software que considera la mantenibilidad [Resultados: a, b, c, d, e, f, g, h, i, j, k]
PTS-04 Registro de trazabilidad [Resultados: j, k]

**Figura 23.**

Prácticas de codificación del modelo MANTUS. [36]

La aplicación de las características del modelo MANTUS se realizó con el propósito de fortalecer la mantenibilidad según la ISO/IEC 25010. Una vez implementadas, los resultados se validaron aplicando la lista de verificación que se presenta en la Figura 24.

		CA	M	CM	CP	R
a)	Las unidades de software cuentan con buenos comentarios.					
b)	Las unidades de software tienen baja complejidad.					
c)	Las unidades de software son fáciles de entender.					
d)	Las unidades de software cuentan con alta facilidad de lectura.					
e)	Las unidades de software son simples.					
f)	Las unidades de software tienen el tamaño adecuado.					
g)	Las unidades de software cuentan con bajo nivel de anidación.					
h)	Las unidades de software tienen un estándar definido.					
i)	Las unidades de software evitan la duplicación.					
j)	Las unidades de software son consistentes.					
k)	Las unidades de software cuentan con trazabilidad.					

**Figura 24.**

Check list para cumplimiento de subcaracterísticas de mantenibilidad. [36]

#### 4.1 Enfoque de Evaluación

La evaluación se ha estructurado bajo un enfoque mixto que combina:

1. **Verificación Estática:** Mediante la revisión del código fuente utilizando las prácticas del modelo MANTuS para asegurar la calidad interna del proyecto realizada durante el desarrollo.
2. **Validación Dinámica:** A través de una validación controlada con desarrolladores externos para medir la calidad del código aplicando tareas de mantenimiento real.

#### 4.2 Objetivos de la Validación

Los objetivos específicos de esta fase de validación son:

- Cuantificar el nivel de adaptación del código fuente a las prácticas de ingeniería de software propuestas por MANTuS.
- Determinar la eficiencia de modificación y la tasa de éxito (TEF) al realizar tareas de mantenimiento evolutivo y correctivo.

- Corroborar que la arquitectura del software facilita la analizabilidad y modificabilidad según los criterios de la ISO/IEC 25010.

### 4.3 Verificación Técnica del Modelo MANTuS

Esta sección presenta la auditoría interna del código, verificando si se aplicaron correctamente las directrices definidas durante la fase de desarrollo.

#### 4.3.1 Cumplimiento de Prácticas de Codificación

A continuación, se detalla la verificación de las 18 prácticas base seleccionadas del modelo MANTuS, agrupadas por su naturaleza técnica:

**DEV-MANT.4.PB1 y PB2: Uso de comentarios y propósitos.** Se hace uso de comentarios a lo largo de la codificación y se describe la función de los métodos empleados.

```
/**
 * Componente ProjectMarkerPopup - Popup de información de proyecto
 * @param {Object} location - Datos del proyecto/ubicación
 * @param {Function} onProjectMarkerClick - Callback para editar cronograma
 * @param {Function} onProjectControlsClick - Callback para ver controles
 * @param {Function} onDocumentsClick - Callback para ver documentos
 * @param {boolean} canEditSchedule - Si puede editar cronogramas
 */
const ProjectMarkerPopup = ({
  location,
  onProjectMarkerClick,
  onProjectControlsClick,
  onDocumentsClick,
  canEditSchedule,
}) => {
```

**Figura 25.**

Evidencia de comentarios descriptivos en el código.

**DEV-MANT.4.PB4: Organización del código.** El código fuente está indentado, de modo que se puede identificar de mejor manera dónde inician o finalizan las instrucciones.

```

<div className="bg-white dark:bg-gray-800 rounded-xl shadow-sm border border-gray-200 dark:border-gray-700 p-4">
  <div className="flex flex-col lg:flex-row items-end gap-3">
    { /* Grupo de Inputs */ }
    <div className="flex-1 grid grid-cols-2 gap-3 w-full">
      <DateInput
        label="Desde"
        value={startDate}
        onChange={(e) => onChange("startDate", e.target.value)}
      />
      <DateInput
        label="Hasta"
        value={endDate}
        min={startDate}
        disabled={!startDate}
        onChange={(e) => onChange("endDate", e.target.value)}
      />
    </div>

    { /* Botón de Acción */ }
    <button
      onClick={onApply}
      disabled={!startDate || !endDate}
      className={`
        flex items-center justify-center gap-2 px-6 h-[38px] rounded-lg font-bold text-sm transition-all duration-200 s
        ${(!startDate || !endDate)
          ? 'bg-gray-100 text-gray-400 cursor-not-allowed border border-gray-200 dark:bg-gray-700 dark:text-gray-500'
          : 'bg-indigo-600 hover:bg-indigo-700 text-white shadow-indigo-200 hover:shadow-md active:transform active:s
        }
      `}
    >
      <FaFilter className={(!startDate && !endDate) ? "animate-pulse" : ""} />
      <span>Filtrar</span>
    </button>
  </div>
</div>

```

Figura 26.

Indentación correcta en componente React.

**DEV-MANT.4.PB5: Asignar nombres claros.** Los nombres de las variables describen claramente a qué hacen referencia.

```

$visit = Visit::create([
  'control_id' => $validatedData['visit_control_id'],
  'scheduled_date' => $validatedData['scheduled_date'],
  'completed_date' => $validatedData['completed_date'],
  'status' => $validatedData['status'],
  'result' => $validatedData['result'],
  'observations' => $validatedData['observations'],
  'report_path' => null,
  'report_original_name' => null,
]);

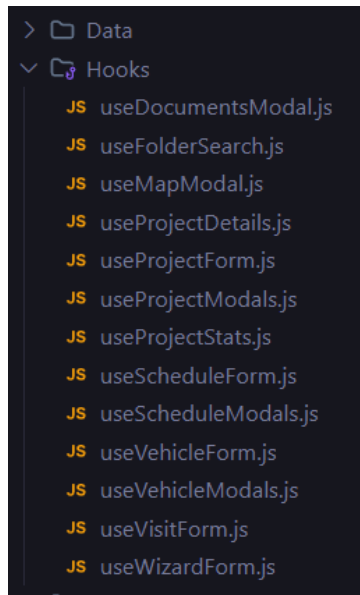
// Guardar archivo usando StorageService si existe
if ($request->hasFile('report_path')) {
  $file = $request->file('report_path');
  $reportPath = $this->storageService->storeFile($file, $visit->visit_id, 'visit_reports');
  $reportOriginalName = $file->getClientOriginalName();
}

```

Figura 27.

Variables con nomenclatura descriptiva.

**DEV-MANT.4.PB6: Controlar abreviaciones.** Se hace uso de abreviaciones únicamente cuando se trata de prefijos estandarizados.



**Figura 28.**

Estructura de archivos sin abreviaciones confusas.

**DEV-MANT.4.PB7: Controlar sentencias.** Se dividen las sentencias largas para mayor facilidad de lectura.

```
$project->agreements()->updateOrCreate(  
  ['agreement_id' => $validatedData['agreement_id'] ?? $agreement->agreement_id ?? null],  
  $agreementData  
);  
  
// --- Lógica de Pago 1 (Actualizar o Crear) ---  
$payData1 = [  
  'amount' => $validatedData['amount'],  
  'pay_date' => $validatedData['pay_date'],  
];  
$firstPay = $project->pays()->where('corresponding_phase', '1')->first();
```

**Figura 29.**

División de sentencias largas en múltiples líneas.

**DEV-MANT.4.PB8: Controlar paréntesis.** Se hace uso correcto de paréntesis para mayor claridad en operaciones lógicas.

```

/**
 * Alternar selección de un checkbox individual
 * @param {number|string} id - ID del proyecto
 */
const handleCheckboxChange = (id) => {
  setSelectedProjects((prev) =>
    prev.includes(id)
      ? prev.filter((item) => item !== id)
      : [...prev, id],
  );
};

```

**Figura 30.**

Uso de paréntesis en lógica condicional.

**DEV-MANT.4.PB9: Determinar funcionalidades.** No se han implementado funcionalidades innecesarias; cada método tiene una sola responsabilidad.

```

public function index()
{
  // La lógica de consulta está en el Modelo
  $projects = Project::getProjectsWithOwner();
  $userWithPermissions = User::getPermissionAuthUser();

  return Inertia::render("Project/Projects", [
    'auth' => ['user' => $userWithPermissions],
    'projects' => $projects,
  ]);
}

```

**Figura 31.**

Métodos con responsabilidad única.

**DEV-MANT.4.PB10: Dividir métodos.** Existen métodos que llaman a otros más pequeños dentro de la misma clase o función.

```
public function createProject(Request $request): Project
{
    $validatedData = $request->validated();

    // Log para debugging
    Log::info('Datos validados para crear proyecto:', $validatedData);

    return DB::transaction(function () use ($request, $validatedData) {

        $owner = $this->findOrCreateOwner($validatedData);
```

Figura 32.

División de lógica compleja en métodos auxiliares.

**DEV-MANT.4.PB11: Controlar tamaño.** El tamaño de las unidades de código es controlado para facilitar su mantenimiento.

```
98
99 // Crear DivIcon de Leaflet con estilos personalizados
100 return L.divIcon({
101     html: `
102         <div style="
103             background-color: ${colorInfo.fillColor};
104             width: 36px;
105             height: 36px;
106             border-radius: 50%;
107             border: 3px solid ${colorInfo.color};
108             display: flex;
109             align-items: center;
110             justify-content: center;
111             box-shadow: 0 3px 8px rgba(0,0,0,0.4);
112             transition: all 0.2s ease;
113         ">
114             ${iconHtml}
115         </div>
116     `,
117     className: 'custom-div-icon',
118     iconSize: [36, 36],
119     iconAnchor: [18, 18],
120     popupAnchor: [0, -18]
121 });
122
```

Figura 33.

Ejemplo de función con tamaño controlado.

**DEV-MANT.4.PB12: Controlar nivel de anidación.** No se usan más de 3 ciclos anidados para evitar complejidad ciclomática alta.

```

1 reference | 0 overrides
public function deleteProjects(string $ids): void
{
    $projectIds = explode(',', $ids);

    DB::transaction(function () use ($projectIds) {
        foreach ($projectIds as $id) {
            $project = Project::findOrFail($id);
            $project->delete(); // Esto dispara el evento 'deleting' del modelo
        }
    });
}

```

**Figura 34.**  
Control de estructuras anidadas.

**DEV-MANT.4.PB13: Evitar duplicación.** Se utilizó componentes para reducir el código duplicado.

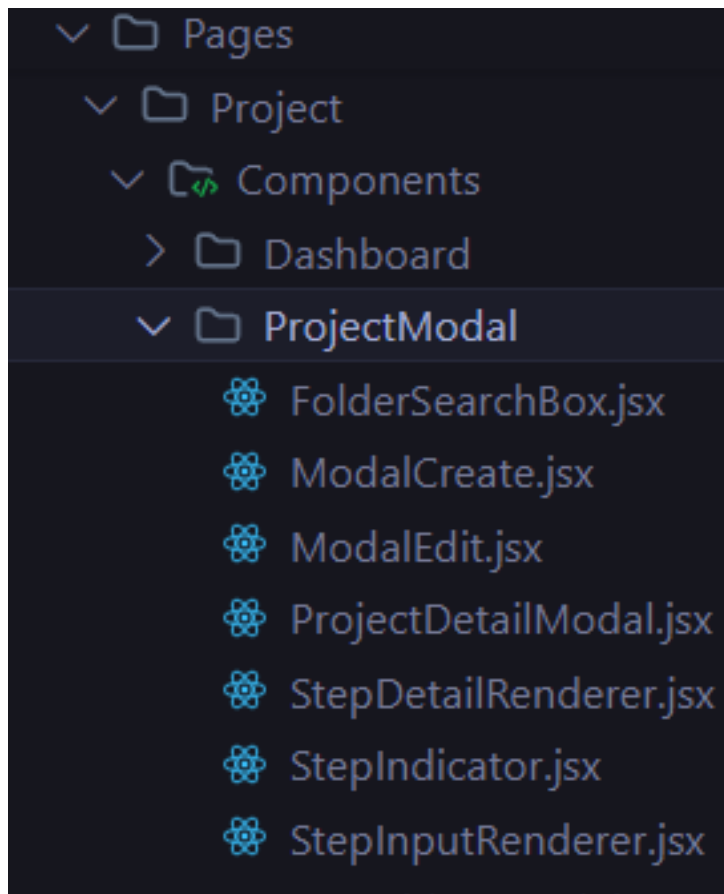
```

/* 4. Grid de detalles principales (Ahora usa el componente reutilizable) */
<div className="grid grid-cols-2 lg:grid-cols-4 gap-3 sm:gap-4 md:gap-6 p-3 sm:p-4 bg-gray-50 dark:bg-gray-900/50 round
  <DetailItem
    icon={
      <FaCalendarAlt className="w-3 h-3 sm:w-4 sm:h-4" />
    }
    label="Fecha Program. any
    value={visit.scheduled_date}
  />
  <DetailItem
    icon={
      <FaCalendarAlt className="w-3 h-3 sm:w-4 sm:h-4" />
    }
    label="Fecha de Cierre"
    value={completedDate}
  />
  <DetailItem
    icon={
      <FaClipboardCheck className="w-3 h-3 sm:w-4 sm:h-4" />
    }
    label="Estado"
    value={visit.status}
  />
  <DetailItem
    icon={
      <FaClipboardCheck className="w-3 h-3 sm:w-4 sm:h-4" />
    }
    label="Resultado"
    value={visit.result}
  />
</div>

```

**Figura 35.**  
Reutilización de lógica para reducir duplicidad.

**DEV-MANT.4.PB15: Utilizar convenciones.** Se maneja un estándar al momento de nombrar paquetes y componentes.



**Figura 36.**

Convenciones de nombres en estructura de directorios.

**DEV-MANT.4.PB16: Controlar tamaño de unidad.** Las piezas de funcionalidad de bajo nivel se mantienen pequeñas.

```
const handleScheduleAction = (e) => {
  e.preventDefault();
  if (onProjectMarkerClick) onProjectMarkerClick(location);
};

const handleDocsAction = (e) => {
  e.preventDefault();
  if (onDocumentsClick) onDocumentsClick(location);
};

const handleControlsAction = (e) => {
  e.preventDefault();
  if (onProjectControlsClick) onProjectControlsClick(location);
};

const colorInfo = getMarkerColorByScheduleStatus(
  location.scheduleStatus,
  location.hasSchedule,
);
```

**Figura 37.**

Funciones manejadoras de eventos (handlers) concisas.

**DEV-MANT.4.PB17 y PB18: Verificar trazabilidad y Actualizar artefactos.** El código refleja fielmente los requisitos y se actualiza ante cambios.

```

/**
 * Actualiza un proyecto existente en la base de datos.
 * La validación se maneja en UpdateProjectRequest.
 * La lógica de negocio se maneja en ProjectService.
 *
 * @param UpdateProjectRequest $request Datos validados del proyecto a actualizar
 * @param Project $project Instancia del proyecto a actualizar
 * @return \Illuminate\Http\RedirectResponse Redirecciona a la lista de proyectos con mensaje de éxito o error
 * @throws \Exception si ocurre un error durante la actualización del proyecto
 */
0 references | 0 overrides
public function update(UpdateProjectRequest $request, Project $project)
{
    try {
        $this->projectService->updateProject($request, $project);
        return redirect()->route('project.index')->with('success', 'Proyecto actualizado exitosamente.');
```

**Figura 38.**  
Manejo de errores y consistencia en actualizaciones.

### 4.3.2 Check List de Subcaracterísticas

Para vincular las prácticas con la norma ISO/IEC 25010, se agruparon los resultados según las subcaracterísticas que impactan (CA: Capacidad de Análisis, M: Modularidad, CM: Capacidad de Modificación, CP: Capacidad de Prueba, R: Reusabilidad).

**Tabla 34.**  
Matriz de cumplimiento por subcaracterística.

Las Unidades de Software...	CA	M	CM	CP	R
Cuentan con buenos comentarios	X	X			
Tienen baja complejidad	X	X	X	X	X
Son fáciles de entender	X		X	X	
Tienen líneas de código fáciles de leer	X		X	X	
Son simples	X	X	X	X	X
Tienen el tamaño adecuado	X	X	X	X	
Cuentan con bajo nivel de anidación	X		X	X	
Tienen un estándar definido	X		X		X
Evitan la duplicación		X	X		X

<b>Las Unidades de Software...</b>	<b>CA</b>	<b>M</b>	<b>CM</b>	<b>CP</b>	<b>R</b>
Son consistentes	X	X	X	X	X
Cuentan con trazabilidad	X		X		

#### 4.4 Validación Experimental de la Modificabilidad

Se diseñó una guía práctica para medir el éxito del proyecto ante modificaciones requeridas, mismas que serán aplicadas por un desarrollador que no está involucrado al proyecto y que posea habilidades en lógica de programación.

##### Procedimiento

Para el desarrollo de la guía práctica se creó una nueva rama en el repositorio a modo de prueba con el propósito de realizar los cambios sin afectar al estado final del proyecto. Previo a la ejecución de la prueba se llevó a cabo una introducción a la arquitectura y propósito del proyecto. Al llevar a cabo las pruebas se cronometró el tiempo en dos periodos: el primero enfocado en el análisis para encontrar los archivos que van a ser modificados, y el segundo enfocado a ejecutar netamente la modificación del código fuente.

##### Criterios de Evaluación

- **Éxito:** Se definió la actividad como exitosa una vez que terminada la modificación de la tarea, el proyecto funciona y no rompe otras funcionalidades.
- **Tiempo:** Se comparó el tiempo que le tomó a los participantes en realizar la modificación contra una estimación de tiempo previamente planteada para cada tarea.

#### 4.4.1 Tareas definidas para la evaluación

Se plantearon 4 tareas de complejidad incremental para evaluar distintos aspectos del sistema y verificar que cumpla con los criterios de analizabilidad, modificabilidad, capacidad de prueba, modularidad y reutilización pertenecientes a la mantenibilidad propuestos por la norma ISO/IEC 25010:

**Tabla 35.**  
Tareas definidas para la evaluación experimental.

<b>Tarea</b>	<b>Requerimiento</b>	<b>Objetivo</b>	<b>Tiempo Est.</b>
<b>TAREA 1:</b> Modificación de Interfaz (Frontend)	El cliente desea cambiar el título principal del Dashboard de proyectos. Actualmente dice “Resumen Ejecutivo de Construcción”. Cámbielo por: “Sistema de Gestión de Construcciones v1.0”.	Evaluar la facilidad para encontrar componentes de vista y textos estáticos.	5 min
<b>TAREA 2:</b> Estilo Visual (Frontend)	En el listado de inspecciones, los iconos de Fecha Programada, Fecha de Cierre, Estado y Resultado están sin color definido. El cliente lo quiere de color Verde, Rojo, Azul y Morado.	Evaluar la modularidad de los estilos y componentes reutilizables.	5 min
<b>TAREA 3:</b> Restricción de Datos (Fullstack)	En el formulario de creación de proyectos, el campo “Observaciones” actualmente es opcional. Hágalo Obligatorio. El sistema debe mostrar un error si se intenta guardar vacío.	Evaluar la facilidad para modificar reglas de negocio obligatorias.	10 min

<b>Tarea</b>	<b>Requerimiento</b>	<b>Objetivo</b>	<b>Tiempo Est.</b>
<b>TAREA 4:</b> Nueva Funcionalidad (Navegación)	Agregue un nuevo botón en la barra de navegación superior (Navbar) que diga “Soporte” y que, al hacer clic, redirija a <a href="https://www.utn.edu.ec">https://www.utn.edu.ec</a> .	Evaluar la estructura del layout principal y la facilidad de expansión.	15 min

#### 4.4.2 Resultados Experimentales

A continuación se presentan los resultados consolidados de la aplicación de las tareas de mantenimiento.

**Tabla 36.**  
Resultados consolidados de la validación experimental.

<b>ID Tarea</b>	<b>Fase de Análisis</b>		<b>Fase de Modificación</b>	
	<b>¿Correcto?</b>	<b>Duración</b>	<b>¿Correcto?</b>	<b>Duración</b>
RP1	SI	00:04:46	SI	00:01:50
RP2	SI	00:06:04	SI	00:02:17
RP3	SI	00:13:24	SI	00:01:41
RP4	SI	00:08:12	SI	00:11:02
<b>Total</b>	<b>4/4</b>	<b>32:26</b>	<b>4/4</b>	<b>16:50</b>

#### 4.4.3 Análisis de Tiempos de Ejecución

Se realizó una comparativa entre los tiempos estimados inicialmente contra el tiempo real promedio que le tomó a el evaluador externo.

**Tabla 37.**  
Comparativa de Tiempos Estimados vs Reales.

<b>Tarea</b>	<b>Tiempo estimado</b>	<b>Tiempo real promedio</b>
RP1	5 minutos	6:36 minutos
RP2	5 minutos	8:21 minutos
RP3	10 minutos	15:05 minutos
RP4	15 minutos	19:14 minutos

#### 4.4.4 Verificación de Resultados

Con el objetivo de validar los resultados de las tareas de mantenimiento, se utilizó el indicador de Tasa de Éxito en el Encuentro de Fallos (TEF). Según Erazo, Florez y Pino, este cálculo permite determinar la efectividad de las modificaciones al código fuente, midiendo la proporción de tareas completadas exitosamente frente a aquellas que presentaron errores críticos o no fueron finalizadas.

$$TEF = 1 - \left(\frac{A}{B}\right)$$

Donde:

- $A$  (Tareas fallidas) = 0
- $B$  (Total tareas intentadas) = 4 tareas

$$TEF = 1 - \left(\frac{0}{4}\right) = 1$$

El resultado de TEF con un valor de 1 indica una mejor capacidad del sistema para encontrar fallos y que las modificaciones se completaron funcionalmente sin introducir nuevos errores.

#### 4.4.5 Interpretación y Discusión de Resultados

Los resultados experimentales validan positivamente la arquitectura del sistema bajo los criterios de la norma ISO/IEC 25010. El hecho de que los tiempos de análisis fueran cercanos a la estimación experta confirma que la curva de aprendizaje del código es baja, permitiendo a nuevos desarrolladores entender la presentación visual y la estructura de componentes rápidamente.

Aunque la implementación de nuevas funcionalidades y el ajuste de reglas de negocio demandaron un tiempo superior al estimado, dada la necesidad de comprender la lógica del Backend, el indicador TEF de 1 es concluyente: el sistema permite realizar cambios complejos y escalables sin introducir fallos críticos. Esto contrasta favorablemente con el estándar de la industria, donde la deuda técnica suele afectar a que un proyecto cambie de forma segura, validando así que la aplicación del modelo MANTuS ha generado un producto de software mantenible.

### 4.5 Síntesis de Resultados

#### 4.5.1 Consolidado de Evaluaciones

El proceso de validación ha permitido triangular la calidad del software desde dos perspectivas complementarias, obteniendo resultados satisfactorios en ambas instancias:

- **Calidad Interna (Verificación Estática):** Se garantizó mediante la auditoría interna del código, cumpliendo con las prácticas base del modelo MANTuS, asegurando un código limpio y estandarizado.
- **Calidad Externa (Validación Dinámica):** Se ejecutó las tareas de mantenimiento, alcanzando una Tasa de Éxito en el Encuentro de Fallos (TEF) de 1, lo que demuestra la robustez del sistema ante cambios futuros.

## 4.5.2 Cumplimiento de Requisitos

Para evidenciar la calidad del producto de software, se presenta la tabla en donde se encuentran los requisitos planteados por parte de INNOVA-EP y el estado de cumplimiento.

**Tabla 38.**  
Matriz de Cumplimiento de Requisitos del Sistema.

<b>Requisito INNOVA-EP</b>	<b>Evidencia de Validación</b>	<b>Estado</b>
<b><i>Requisitos Funcionales</i></b>		
RF1: Gestión de Convenios y Cronogramas	Criterios de Aceptación (HU-02)	Cumple
RF2: Asignación Recursos	Criterios de Aceptación (HU-05)	Cumple
RF3: Registro de Inspecciones por Fase	Criterios de Aceptación (HU-04, HU-06, HU-07)	Cumple
RF4: Geolocalización	Criterios de Aceptación (HU-08)	Cumple
RF5: Gestión de Pagos	Criterios de Aceptación (HU-02)	Cumple
RF6: Control Paralizaciones de Obras	Criterios de Aceptación (HU-02, HU-06)	Cumple
RF7: Reportes Ejecutivos	Criterios de Aceptación (HU-09 y HU-10)	Cumple
<b><i>Requisitos No Funcionales</i></b>		
RNF1: Accesibilidad Móvil	Criterios de Aceptación (HU-06)	Cumple
RNF2: Seguridad Info.	Criterios de Aceptación (HU-01)	Cumple
RNF3: Mantenibilidad	Criterios de Aceptación y Auditoría MANTuS (HU-11)	Cumple
RNF4: Integridad Datos	Criterios de Aceptación (HU-02)	Cumple
RNF5: Interoperabilidad	Criterios de Aceptación (HU-03)	Cumple

## CONCLUSIONES

Durante la primera sección se definió el uso de una arquitectura REST desacoplandola en servicios y alineándolos a las necesidades para el control de proyectos de construcción de INNOVA-EP junto con los estándares para el desarrollo Web. Al utilizar este enfoque se estableció a la norma ISO/IEC 25010 como el eje central para obtener un producto de software de calidad, garantizando no solo la funcionalidad, sino la mantenibilidad del sistema a largo plazo.

El desarrollo del sistema se realizó separando la lógica de negocio de la interfaz de usuario, utilizando Laravel y React, aplicando principios enfocados en la modularidad y la reutilización de código. El haber aplicado la metodología Scrum facilitó obtener valor al final de cada Sprint por medio de los entregables de software, permitiendo de esta forma verificar las funcionalidades críticas como la georreferenciación y la generación de reportes de las inspecciones, asegurando que el producto final cumpliera con los requisitos para el control de proyectos de construcción.

La validación final del software se realizó de forma interna y externa, para lo cual se aplicó el Modelo MANTUS, comprobando el cumplimiento de buenas prácticas de codificación y documentación. Al realizar las pruebas de mantenibilidad se obtuvo una Tasa de Éxito en el Encuentro de Fallos (TEF) del 100 % (1.0), mostrando que el código fuente es altamente legible y modificable. Estos resultados confirman que el sistema tiene una estructura sólida, permitiendo a la institución adaptar el sistema ante futuros cambios normativos sin que se genere una alta deuda técnica.

## RECOMENDACIONES

Para asegurar que el sistema cumpla con las nuevas necesidades de los usuarios, se recomienda realizar actualizaciones periódicas basadas en la información de los inspectores de campo durante su uso, priorizando la optimización de la interfaz en dispositivos móviles. De igual forma, debido a la evolución rápida de la tecnología web, se debe mantener actualizadas las dependencias de Laravel y las librerías de React para evitar brechas de seguridad y aprovechar mejoras de rendimiento.

De acorde a como aumente el uso del sistema, se recomienda mantener una escalabilidad, que permita separar el servicio de base de datos y el almacenamiento de archivos en entornos independientes, para mejorar los tiempos de respuesta y garantizar que el sistema este disponible durante jornadas de trabajo extensas.

Para garantizar la sostenibilidad del sistema en el aspecto técnico, se recomienda a la empresa INNOVA-EP adoptar formalmente las prácticas del modelo MANTUS como un estándar interno para la calidad del software. Ya que esto apoya a la facilidad para soportes y actualizaciones futuras, así como el agregar nuevos módulos para el sistema.

# Referencias

- [1] G. M. I. Pincay, E. J. Q. Chavéz, J. A. C. Alcívar y V. A. L. Calle, “Gestión de riesgos en planificación de obras civiles: mitigación de retrasos y sobrecostos en construcción, un análisis textual discursivo,” *Revista Ingenio global*, vol. 4, págs. 160-174, 1 feb. de 2025, ISSN: 3028-8800. DOI: 10.62943/rig.v4n1.2025.203. dirección: <https://editorialinnova.com/index.php/rig/article/view/203>.
- [2] M. D. M. Serrano, “Sistema de planificación y monitorización de proyectos de obras civiles,” Universidad Tecnológica Centroamericana, inf. téc., abr. de 2021. dirección: <https://repositorio.unitec.edu/server/api/core/bitstreams/6fcd1ace-8656-4518-b7ab-9bd97749600d/content>.
- [3] J. C. F. ALDANA, “COMPARATIVA DE SOFTWARE DEDICADO A LA GESTIÓN DE PROYECTOS,” 2022. dirección: <https://repository.ucatolica.edu.co/entities/publication/56430218-0c04-44ff-80ae-4a3e635804b3>.
- [4] A. F. S. Alarcón y W. E. S. Lozano, “Implementación de un prototipo de software para el control de inicio y avance en las tareas de las obras de construcción,” 2023. dirección: <https://repository.udistrital.edu.co/server/api/core/bitstreams/3e17f06e-54d7-4980-b590-0b15895e372b/content>.
- [5] P. S. M. Montenegro, “DESARROLLO DEL MÓDULO DE PLANTILLAS PARA EL SISTEMA INTEGRADO DE ACTIVIDAD DOCENTE (SIAD) DE LA CARRERA DE SOFTWARE DE LA UNIVERSIDAD TÉCNICA DEL NORTE, APLICANDO EL ESTÁNDAR ISO/IEC 25010 PARA MANTENIBILIDAD,” Universidad Técnica del Norte, inf. téc., 2021. dirección: <https://repositorio.utn.edu.ec/handle/123456789/11745>.
- [6] Naciones Unidas, *Objetivos y metas de desarrollo sostenible - Objetivo 9: Industria, innovación e infraestructura*, Accedido: 2024-05-14, 2024. dirección: <https://www.un.org/sustainabledevelopment/es/infrastructure/>.
- [7] Secretaría Nacional de Planificación, *Plan de Creación de Oportunidades 2021-2025*, Gobierno del Ecuador, 2021. dirección: <https://www.planificacion.gob.ec/wp->

content/uploads/2021/09/Plan-de-Creacio%CC%81n-de-Oportunidades-2021-2025-Aprobado.pdf.

- [8] K. K. Najj, M. Gunduz, F. H. Alhenzab, H. Al-Hababi y A. H. Al-Qahtani, “A Systematic Review of the Digital Transformation of the Building Construction Industry,” *IEEE Access*, vol. 12, págs. 31 461-31 487, 2024, ISSN: 21693536. DOI: 10 . 1109 / ACCESS . 2024 . 3365934.
- [9] H. Rojas, A. Yaguana y F. Baculima, “HERRAMIENTA PARA REVISIÓN DIGITAL Y CAPTACIÓN DE DATOS DE PROYECTOS DE CONSTRUCCIÓN,” vol. 9, 3 2022, ISSN: 1390-9592. dirección: <http://portal.amelica.org/ameli/journal/367/3673657005/>.
- [10] S. Eugenia et al., “Plan de desarrollo e implementación de un sistema para el monitoreo y control de los proyectos en ejecución de la empresa S&S Constructora,” inf. téc., 2024. dirección: <https://repositorio.uide.edu.ec/handle/37000/7254>.
- [11] V. A. V. Herrera, “DESARROLLO DE UN SOFTWARE PROGRAMADO EN MATLAB CON UNA BASE DE DATOS, QUE PERMITA GENERAR PRECIOS UNITARIOS, PRESUPUESTOS, REAJUSTE, CRONOGRAMA VALORADO DE TRABAJO Y LAS ESPECIFICACIONES TÉCNICAS DE CADA RUBRODE INGENIERO CIVIL,” UNIVERSIDAD TÉCNICA DE AMBATO, inf. téc., mar. de 2021. dirección: <https://repositorio.uta.edu.ec/handle/123456789/32497>.
- [12] G. G. P. Majano, “Aplicación del método de Diseño Generativo en el sector de la construcción en el marco de los Objetivos de Desarrollo Sostenible,” *Gaceta Técnica*, vol. 26, págs. 72-88, 1 dic. de 2024, ISSN: 2477-9539. DOI: 10 . 51372 / gacetatecnica261 . 5. dirección: <https://revistas.uclave.org/index.php/gt/article/view/5111>.
- [13] E. Á. Padilla y E. A. Cabot, “La investigación en proyectos: alternativa de metodología para solucionar problemas en escenarios sociales,” *Revista Varela*, vol. 21, págs. 1-9, 2021. dirección: <https://revistavarela.uclv.edu.cu/index.php/rv/article/view/100>.

- [14] R. Hussamadin, G. Jansson y J. Mikkavaara, “Digital Quality Control System—A Tool for Reliable On-Site Inspection and Documentation,” *Buildings*, vol. 13, 2 feb. de 2023, ISSN: 20755309. DOI: 10.3390/buildings13020358. dirección: <https://doi.org/10.3390/buildings13020358>.
- [15] C. T. M. Eduardo e I. V. A. Jorge, “Identificación de las principales causas de riesgo en la gestión de proyectos de construcción en Ecuador,” Tesis doct., UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL, 2020. dirección: <http://repositorio.ucsg.edu.ec/handle/3317/14454>.
- [16] Ministerio de Desarrollo Urbano y Vivienda, *Documentos Normativos NEC – Norma Ecuatoriana de la Construcción*, Gobierno del Ecuador, 2024. dirección: <https://www.habitatyvivienda.gob.ec/documentos-normativos-nec-norma-ecuatoriana-de-la-construccion/>.
- [17] C. Astudillo, “ANÁLISIS DEL COMPORTAMIENTO NO LINEAL DE UNA ESTRUCTURA DE HORMIGÓN ARMADO DE 4 PLANTAS CONSTRUIDA CON NORMA NEC 2000 Y 2015, UBICADO EN EL BARRIO FORESTAL MEDIA, PARRROQUIA LA FERROVIARIA, CANTÓN QUITO, PROVINCIA PICHINCHA 2022 - 2023,” Tesis doct., UNIVERSIDAD POLITÉCNICA SALESIANA, 2023. dirección: <https://dspace.ups.edu.ec/bitstream/123456789/25251/1/TTS1329.pdf>.
- [18] R. Hernández, C. Alejandro, Y. Jaramillo y V. Israel, “DISEÑO DE UNA EDIFICACIÓN PARA USO RESIDENCIAL DE ACERO ESTRUCTURAL SISMORRESISTENTE MEDIANTE LA COMPARACIÓN DE TÉCNICAS CONSTRUCTIVAS EN LAS JUNTAS VIGA-COLUMNA,” Tesis doct., UNIVERSIDAD POLITÉCNICA SALESIANA, 2024. dirección: <https://dspace.ups.edu.ec/bitstream/123456789/29491/1/MSQ934.pdf>.
- [19] C. Aguilar, G. Juana, R. Sierra, L. Estrella, S. Manchego y V. Hugo, “Propuesta de mejora para el proceso de importación de equipos de telecomunicaciones de una empresa integradora de TI, año 2024,” Tesis doct., Universidad Peruana de Ciencias Aplicadas (UPC), nov. de 2024. dirección: <https://repositorioacademico.upc>.

edu . pe / bitstream / handle / 10757 / 682756 / Ccama \_ AG . pdf ; jsessionid = 8E844DF72ACFBA5BADEB29A89B019C5C?sequence=1.

- [20] J. P. C. Gomezjurado y M. E. V. Duffau, “Gestión de procesos de negocio como mecanismo de transparencia y Gobierno abierto en entidades públicas de Ecuador entre 2016-2020,” *Estado & comunes, revista de políticas y problemas públicos*, vol. 1, 14 ene. de 2022, ISSN: 1390-8081. DOI: 10.37228/estado\_comunes.v1.n14.2022.249. dirección: [https://revistas.iaen.edu.ec/index.php/estado\\_comunes/article/view/249/449](https://revistas.iaen.edu.ec/index.php/estado_comunes/article/view/249/449).
- [21] K. Das, S. Khursheed y V. K. Paul, “The impact of BIM on project time and cost: insights from case studies,” *Discover Materials*, vol. 5, 1 dic. de 2025, ISSN: 27307727. DOI: 10.1007/s43939-025-00200-2.
- [22] M. Eduardo y M. Lían, “CREACIÓN DE APLICACIÓN WEB Y MOBILE PARA ADMINISTRACIÓN DE RESERVAS DE AULAS Y LABORATORIOS PARA LA UNIVERSIDAD SALESIANA,” Tesis doct., UNIVERSIDAD POLITÉCNICA SALESIANA, 2024. dirección: <https://dspace.ups.edu.ec/bitstream/123456789/27864/1/UPS-GT005355.pdf>.
- [23] C. E. G. Gonzales, “UNA REVISIÓN DE LOS PATRONES DE DISEÑO DE SOFTWARE APLICADO A LAS APLICACIONES WEB,” Tesis doct., Universidad Señor de Sipán, 2020. dirección: <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/6783/Gonzales%20Gonzales%20Christian%20Erick.pdf?sequence=1&isAllowed=y>.
- [24] S. C. Ferrández, “Aplicación web con React y Laravel para plataforma académica en tiempo real,” Tesis doct., Universidad Miguel Hernández de Elche, 2022. dirección: <https://dspace.umh.es/bitstream/11000/28555/1/TFG-Calder%C3%B3n%20Ferr%C3%A1ndez,%20Sergio.pdf>.
- [25] M. Olmedo y A. Laura, “DISEÑO DE UN MODELO DE GESTIÓN DE PROYECTOS TECNOLÓGICOS EN CAPGEMINI MÉXICO,” Tesis doct., UNIVERSIDAD

- AUTÓNOMA DEL ESTADO DE HIDALGO, ago. de 2024. dirección: <http://dgsa.uaeh.edu.mx:8080/jspui/bitstream/231104/5534/1/ATD504.pdf>.
- [26] A. : Nicolás et al., “Enseñanza de Métodos ágiles de Desarrollo de Software en Argentina. Estado del Arte,” Tesis doct., UNIVERSIDAD NACIONAL DE LA PLATA, 2020. dirección: [https://sedici.unlp.edu.ar/bitstream/handle/10915/101167/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](https://sedici.unlp.edu.ar/bitstream/handle/10915/101167/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y).
- [27] E. Peters y G. K. Aggrey, “An ISO 25010 based quality model for ERP systems,” *Advances in Science, Technology and Engineering Systems*, vol. 5, págs. 578-583, 2 abr. de 2020, ISSN: 24156698. DOI: 10.25046/aj050272. dirección: <https://www.astesj.com/v05/i02/p72/>.
- [28] A. F. Cordeiro, L. F. Silva y E. Oliveira Jr, “Maintainability metrics for PLA evaluation based on ISO/IEC 25010,” en *UML-Based Software Product Line Engineering with SMarty*. Springer International Publishing, feb. de 2023, págs. 193-218, ISBN: 9783031185564. DOI: 10.1007/978-3-031-18556-4\_9. dirección: [https://link.springer.com/chapter/10.1007/978-3-031-18556-4\\_9](https://link.springer.com/chapter/10.1007/978-3-031-18556-4_9).
- [29] J. Gabriel, B. Ibujés, M. Xavier y M. R. Peñafiel, “Implementación del módulo web ”Gestión de Adquisiciones” para reforzar los procesos administrativos dentro del Sistema Integrado SIGEERN de EMELNORTE S.A., aplicando la característica de mantenibilidad del estándar ISO/IEC 25010.” Tesis doct., Universidad Técnica del Norte, 2023, ISBN: 0967831156. dirección: <https://repositorio.utn.edu.ec/bitstream/123456789/13750/2/04%20ISC%20668%20TRABAJO%20GRADO.pdf>.
- [30] Y. Gao, “Design And Implementation of Construction Project Management System Based on Web Technology,” European Alliance for Innovation n.o., ene. de 2024. DOI: 10.4108/eai.3-11-2023.2342165. dirección: <https://eudl.eu/doi/10.4108/eai.3-11-2023.2342165>.
- [31] M. R. Abdurrahim, “Development Of Website-Based Management Information Systems for Building Construction,” *Ascarya: Journal of Islamic Science, Culture, and Social Studies*, vol. 2, págs. 116-126, 2 sep. de 2022, ISSN: 2798-5083. DOI: 10.53754/

- iscs.v2i2.452. dirección: [https://www.researchgate.net/publication/363840193\\_Development\\_Of\\_Website-Based\\_Management\\_Information\\_Systems\\_for\\_Building\\_Construction](https://www.researchgate.net/publication/363840193_Development_Of_Website-Based_Management_Information_Systems_for_Building_Construction).
- [32] F. Nurunnisa, M. L. Hamzah, Angraini y E. Saputra, “Analyzing the Quality of Web-Based Scholarship Information System Using ISO/IEC 25010 Standard,” en *2024 International Conference on Circuit, Systems and Communication, ICCSC 2024*, Institute of Electrical y Electronics Engineers Inc., 2024, ISBN: 9798350365306. DOI: 10.1109/ICCSC62074.2024.10617354. dirección: <https://ieeexplore.ieee.org/document/10617354>.
- [33] H. S. Suparto y R. H. Dai, “Evaluasi Kualitas Sistem Informasi Pengukuran Prestasi Kerja Berdasarkan ISO/IEC 25010,” *Jambura Journal of Informatics*, vol. 3, págs. 109-120, 2 nov. de 2021, ISSN: 2656-467X. DOI: 10.37905/jji.v3i2.11744. dirección: <https://ejurnal.ung.ac.id/index.php/jji/article/view/11744>.
- [34] J. QAZIMI, F. HALILI, E. RUFATI y G. XHAFERI, “WEB-BASED MANAGEMENT SYSTEM IN TEXTUAL FORMAT FOR OFFERS, CONTRACT AND PAYMENT FOR CONSTRUCTION COMPANY,” *Journal of Natural Sciences and Mathematics of UT-JNSM*, vol. 9, págs. 361-371, 17-18 oct. de 2024, ISSN: 25454072. DOI: 10.62792/ut.jnsm.v9.i17-18.p2833. dirección: <https://journals.unite.edu.mk/Abstract?AIId=1209&DIId=2833>.
- [35] R. K. Mallidi y M. Sharma, “Study on Agile Story Point Estimation Techniques and Challenges,” *International Journal of Computer Applications*, vol. 174, n.º 13, págs. 30-34, ene. de 2021, ISSN: 0975-8887. dirección: [https://www.researchgate.net/publication/348545576\\_Study\\_on\\_Agile\\_Story\\_Point\\_Estimation\\_Techniques\\_and\\_Challenges](https://www.researchgate.net/publication/348545576_Study_on_Agile_Story_Point_Estimation_Techniques_and_Challenges).
- [36] J. E. Martínez, A. F. Gómez y F. J. Pino, “Generando productos software mantenibles desde el proceso de desarrollo: El modelo de referencia MANTuS Creating maintainable software products from the development process: The reference model MANTuS,” *inf. téc.*, 2016, págs. 420-434. dirección: <https://www.researchgate.net/>

publication/305311974\_Generando\_productos\_software\_mantenibles\_  
desde\_el\_proceso\_de\_desarrollo\_El\_modelo\_de\_referencia\_MANTuS/  
fulltext/5787c8e108ae95560407b696/Generando-productos-software-  
mantenibles-desde-el-proceso-de-desarrollo-El-modelo-de-referencia-  
MANTuS.pdf?origin=publication\_detail&\_tp=eyJjb250ZXh0Ijp7ImZpcnNOUGFnZSI6InB1Ymx  
\_\_cf\_chl\_rt\_tk=rrcyXZZacmSOCvSJifRjYUZ3815XXu3lLNBL8r82ILM-1762641649-  
1.0.1.1-WUGdeFYhJ4rbbT0dllyQmGhA1.FgpSuwDOKQBqBFt2w.

## ANEXOS

