

# UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

## Carrera de Software

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA DE SOFTWARE PRESENTADO ANTE LA NOBLE  
UNIVERSIDAD TÉCNICA DEL NORTE.

**“Desarrollo de una aplicación web para la gestión del cementerio Indígena de  
Otavalo de la Organización UNORICO SAMASHUNCHIK utilizando el  
Framework Laravel y aplicando el marco de trabajo Scrum”**



### AUTOR:

Verónica Cecilia Saransig Albancando

### DIRECTOR:

Msc. Xavier Mauricio Rea Peñafiel

Ibarra-Ecuador

2026

**UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA**

**IDENTIFICACIÓN DE LA OBRA**

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

<b>DATOS DE CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	1050193265		
<b>APELLIDOS Y NOMBRES:</b>	Saransig Albancando Verónica Cecilia		
<b>DIRECCIÓN:</b>	Otavalo, Calle Quiroga y Atahualpa cerca al parque La Florida		
<b>EMAIL:</b>	<a href="mailto:vcsaransig@utn.edu.ec">vcsaransig@utn.edu.ec</a> / <a href="mailto:cecilia2002n@gmail.com">cecilia2002n@gmail.com</a>		
<b>TELÉFONO FIJO:</b>	062923549	<b>TELÉFONO MÓVIL:</b>	0990382254

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	Desarrollo de una aplicación web para la gestión del cementerio Indígena de Otavalo de la Organización UNORICO SAMASHUNCHIK utilizando el Framework Laravel y aplicando el marco de trabajo Scrum
<b>AUTOR:</b>	Verónica Cecilia Saransig Albancando
<b>FECHA DE APROBACIÓN:</b>	30/03/2026
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
<b>TITULO POR EL QUE OPTA:</b>	Ingeniería en Software
<b>DIRECTOR:</b>	Ing. Xavier Mauricio Rea Peñafiel, Msc.

## **AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Verónica Cecilia Saransig Albancando, con cédula de identidad Nro. 105019326-5, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 30 días del mes de marzo de 2026

**EL AUTOR:**

Verónica Cecilia Saransig Albancando  
CI: 1050193265

## **CONSTANCIAS**

El autor manifiesta 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 el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 30 días del mes de marzo de 2026

### **EL AUTOR:**

Verónica Cecilia Saransig Albancando  
CI: 1050193265

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

Ibarra 30 de marzo de 2026

Ing. Xavier Mauricio Rea Peñafiel, M

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.

Ing. Xavier Mauricio Rea Peñafiel, MSc.

Director de Trabajo de Grado

## DEDICATORIA

A mis padres, Luis Saransig y Ana Albancando, por su apoyo constante en mi camino y mi mayor fuente de inspiración. Gracias por su amor incondicional, por cada sacrificio silencioso que hicieron por mí para brindarme un futuro mejor. Su apoyo constante y su fe inquebrantable han sido la base que me impulso a seguir adelante y alcanzar este éxito. Este logro, más que mío, es de ustedes.

A mi abuelita, Margarita Cotacachi, con todo mi cariño. Tu amor incondicional y tus consejos han sido mi guía en cada paso. Gracias por estar siempre ahí para mí, por apoyarme y hacerme sentir especial.

A mi tía, Elena Albancando, con su cariño y compañía constante, gracias por estar a mi lado en cada etapa de este proceso, por tu apoyo incondicional, por escuchar mis dificultades y festejar cada éxito. Tu confianza absoluta en mis capacidades fue esencial para culminar este trabajo.

A mi hermana, Lizbeth Saransig, mi confidente y mi compañera incondicional de vida. Gracias por soportar mis momentos más complicados, por compartir conmigo risas en medio del caos y por ser mi refugio emocional durante todos estos años de carrera. Mas que una hermana, eres una gran amiga y me enorgullece profundamente compartir este logro contigo.

A Jordi, por ser mucho más que mi apoyo; por ser mi motivación constante y mi paz en medio del caos. Gracias por tu paciencia infinita, por creer en mi incluso cuando yo misma dudaba, y por llenar mis días de amor y tranquilidad.

A mis amistades, en especial a Lizbeth que ha sido una amiga incondicional y mi gran compañera durante esta etapa universitaria. Gracias por las risas, por los interminables desvelos y no dejarme caer cuando el cansancio pesaba. A todos mis demás amigos y compañeros, gracias por hacer de este camino una experiencia inolvidable.

## AGRADECIMIENTO

A la Universidad Técnica del Norte y a la Facultad de Ingeniería en Ciencias Aplicadas, mi hogar de estudios, por haberme abiertos sus puertas y brindarme una formación de excelencia. Gracias por ser el pilar de mi crecimiento profesional en el desarrollo de software y por brindarme los recursos necesarios para hacer frente a los retos tecnológicos.

A mis docentes, por enseñarme a tener una visión sobre mi profesión, ayudándome a no fijarme en un solo camino, sino en todas las posibilidades. Gracias por compartir su gran experiencia y explicar con claridad conceptos y practicas complejas a lo largo de mi formación.

Un agradecimiento muy especial a mi directo, el MSc. Mauricio Xavier Rea, por indicarme el camino correcto cuando tuve dificultades o dudas y por apoyarme en las etapas difíciles del desarrollo de este trabajo. Sin sus consejos, su paciencia y su guía constante, no hubiese sido posible completar este trabajo con éxito.

Extiendo mi más profundo agradecimiento a mi asesor, MSc. MacArthur Ortega, por su excelente disposición y sus revisiones minuciosas, fueron fundamentales para pulir los detalles de la aplicación.

Finalmente, agradezco a la directiva de la organización UNORICO SAMASHUNCHIK, por brindarme la confianza, apertura y oportunidad de desarrollar esta aplicación para la gestión del cementerio Indígena de Otavalo.

## TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
1.1 Tema .....	1
1.2 Planteamiento del problema .....	1
1.3 Objetivos.....	2
1.3.1 Objetivo General.....	2
1.3.2 Objetivos Específicos .....	2
1.4 Alcance .....	2
1.5 Metodología.....	4
1.6 Justificación .....	5
1.7 Antecedentes.....	6
CAPÍTULO I - MARCO TEÓRICO.....	8
2.1 Gestión de Cementerios.....	8
2.1.1 Procesos de administración .....	8
2.1.2 Problemas en la gestión manual .....	8
2.1.3 Beneficios de la digitalización en cementerios.....	9
2.1.4 Aplicaciones web para la gestión de cementerios .....	10
2.2 Tecnologías de desarrollo.....	10
2.2.1 Lenguaje de programación PHP .....	10
2.2.2 Arquitectura MVC .....	12
2.2.3 Framework Laravel.....	13
2.2.4 Base de datos PostgreSQL.....	14
2.3 Metodología SCRUM.....	15
2.3.1 Principios de la Metodología.....	15
2.3.2 Etapa 1.-Pre-Juego.....	16
2.3.3 Etapa 2.-Juego .....	17
2.3.4 Etapa 3.-Post-Juego .....	17
2.4 Modelo de Éxito y Sistemas de Información de DeLone y McLean.....	18
2.4.1 Aspectos fundamentales del Modelo.....	18
2.4.2 Dimensiones claves para evaluación de aplicaciones: calidad del sistema, uso, satisfacción y beneficios .....	19
2.4.3 Métodos de recopilación de datos .....	21
2.4.4 Interpretación de resultados.....	22
2.5 Trabajos Relacionados.....	23
CAPÍTULO II - DESARROLLO.....	28

3.1	Desarrollo Etapa Pre-Juego .....	28
3.1.1	Definición del equipo SCRUM .....	28
3.1.2	Elicitación de Requerimientos .....	29
3.1.3	Product Backlog .....	51
3.2	Desarrollo Etapa Juego .....	52
3.2.1	Sprint 0 .....	52
3.2.2	Sprint 1 .....	62
3.2.3	Sprint 2 .....	75
3.2.4	Sprint 3 .....	87
3.2.5	Sprint 4 .....	107
3.3	Desarrollo Etapa Pos-Juego .....	119
3.3.1	Despliegue del Producto .....	120
3.3.2	Entrega del Producto .....	120
	CAPÍTULO III .....	121
4.1	Herramientas de Evaluación .....	121
4.1.1	Modelo de evaluación seleccionado .....	121
4.1.2	Delimitación de la evaluación .....	121
4.1.3	Matriz Operacional .....	121
4.2	Procesamiento de datos .....	124
4.2.1	Recolección de datos .....	124
4.2.2	Verificación por estadística descriptiva.....	124
4.3	Interpretación de datos.....	127
4.3.1	Dimensiones del modelo DeLone & McLean .....	127
4.4	Análisis de resultados .....	142
	Conclusiones y recomendaciones.....	143
	Conclusiones .....	143
	Recomendaciones .....	144
	Bibliografía.....	145
	Anexos.....	151
6.1	Anexo A. Fotografías de la encuesta Linkert usando el modelo DeLone and McLean .....	151
6.2	Anexo E. Certificado de Entrega de la Aplicación Web (Organización UNORICO SAMASHUNCHIK).....	152

## ÍNDICE DE FIGURAS

Figura 1. Árbol de problemas .....	2
Figura 2. Arquitectura a desarrollar.....	4
Figura 3. Patrón de diseño MVC.....	13
Figura 4. Dimensiones del Modelo DeLone and McLean.....	21
Figura 5. Fases de desarrollo de Scrum.....	28
Figura 6 Arquitectura General del Proyecto.....	53
Figura 7 Estructura General del Proyecto.....	56
Figura 8 Diagrama Entidad Relación de la Base de Datos.....	57
Figura 9 Iniciar Sesión.....	64
Figura 10 Credenciales Incorrectas .....	65
Figura 11 Dashboard (Ingreso Exitoso) .....	65
Figura 12 ¿Olvidaste tu contraseña?.....	66
Figura 13 Mi perfil .....	66
Figura 14 Gestión Usuarios .....	67
Figura 15 Gestión Roles .....	68
Figura 16 Gestión Permisos.....	68
Figura 17 Gestión de Cantones.....	69
Figura 18 Gestión de Parroquias .....	69
Figura 19 Gestión de comunidades u organizaciones .....	70
Figura 20 Gestión de Servicios.....	77
Figura 21 Gestión de Beneficios .....	78
Figura 22 Gestión de Socios.....	78
Figura 23 Nuevo Socio .....	79
Figura 24 Editar / Actualizar Socio .....	79
Figura 25 Detalle del Socio .....	80
Figura 26 Eliminar Socio.....	80
Figura 27 Buscar Socio en tiempo real.....	81
Figura 28 Control de Socios Exonerados .....	82
Figura 29 Descargar Reporte Socios PDF y Excel.....	82
Figura 30 Reporte Socios PDF .....	83
Figura 31 Reporte Socios Excel .....	83
Figura 32 Gestión Bloques .....	90
Figura 33 Crear Bloque pestaña Datos .....	90
Figura 34 Crear Bloque pestaña ubicación GIS .....	91
Figura 35 Gestión Nichos.....	92
Figura 36 Crear Nicho pestaña Ubicación.....	92
Figura 37 Crear Nicho pestaña Datos Técnicos .....	93
Figura 38 Código QR .....	93
Figura 39 Búsqueda de nichos.....	94
Figura 40 Botón Ubicar Mapa .....	95
Figura 41 Mapa del Catastro con ubicación de un nicho con asignación.....	95
Figura 42 Gestión Fallecidos.....	96
Figura 43 Crear Fallecidos con datos autocompletados pestaña personal.....	96
Figura 44 Crear fallecido con datos autocompletados pestaña detalles y notas.....	97
Figura 45 Registro nuevo fallecido .....	97

Figura 46 Cambio de estatus a "Fallecido" .....	98
Figura 47 Buscar fallecido en tiempo real.....	98
Figura 48 Gestión Asignaciones o Inhumación.....	99
Figura 49 Crear nueva Asignación pestaña Nicho .....	100
Figura 50 Crear nueva Asignación pestaña Socio .....	100
Figura 51 Crear nueva Asignación pestaña Fallecido .....	101
Figura 52 Exhumar Fallecido .....	101
Figura 53 Confirmar Exhumación.....	102
Figura 54 Generar Reportes.....	102
Figura 55 Generar Reporte PDF/Excel.....	103
Figura 56 Gestión Historial de Pagos .....	108
Figura 57 Verificar años cancelados o pendientes .....	109
Figura 58 Historial de Pago de un Socio .....	109
Figura 59 Gestión de Facturas .....	110
Figura 60 Crear nueva Factura .....	110
Figura 61 Emitir Factura .....	111
Figura 62 Anular Factura.....	111
Figura 63 Imprimir Factura .....	112
Figura 64 Registrar nuevo pago.....	112
Figura 65 Años seleccionados a cancelar o pagar .....	113
Figura 66 Nuevo registro insertado .....	113
Figura 67 Auditoría .....	114
Figura 68 Búsqueda en tiempo real (nombre usuario) .....	114
Figura 69 Calidad del Sistema. Pregunta 1. [Elaboración propia] .....	128
Figura 70 Calidad del Sistema. Pregunta 2. [Elaboración propia] .....	128
Figura 71 Sistema de Calidad. Pregunta 3. [Elaboración propia] .....	129
Figura 72 Sistema de Calidad. Pregunta 4. [Elaboración propia] .....	129
Figura 73 Calidad de la Información. Pregunta 5. [Elaboración propia] .....	130
Figura 74 Calidad de la Información. Pregunta 6. [Elaboración propia] .....	131
Figura 75 Calidad de la Información. Pregunta 7. [Elaboración propia] .....	131
Figura 76 Calidad de la Información. Pregunta 8. [Elaboración propia] .....	132
Figura 77 Calidad de Servicio. Pregunta 9. [Elaboración propia].....	133
Figura 78 Calidad del Servicio. Pregunta 10. [Elaboración propia].....	133
Figura 79 Calidad del Servicio. Pregunta 11. [Elaboración propia].....	134
Figura 80 Uso-Intención de Uso. Pregunta 12. [Elaboración propia] .....	135
Figura 81 Uso-Intención de Uso. Pregunta 13. [Elaboración propia] .....	135
Figura 82 Uso-Intención de Uso. Pregunta 14. [Elaboración propia] .....	136
Figura 83 Satisfacción del Usuario. Pregunta 15. [Elaboración propia] .....	137
Figura 84 Satisfacción del Usuario. Pregunta 16. [Elaboración propia] .....	137
Figura 85 Satisfacción del Usuario. Pregunta 17. [Elaboración propia] .....	138
Figura 86 Beneficios Netos. Pregunta 18. [Elaboración propia].....	139
Figura 87 Beneficios Netos. Pregunta 19. [Elaboración propia].....	139
Figura 88 Beneficios Netos. Pregunta 20. [Elaboración propia].....	140
Figura 89 Beneficios Netos. Pregunta 21. [Elaboración propia].....	140
Figura 90 Beneficios Netos. Pregunta 22. [Elaboración propia].....	141

## ÍNDICE DE TABLAS

Tabla 1. Antecedentes.....	6
Tabla 2. Roles del Proyecto.....	28
Tabla 3 Historia de Usuario 1.....	29
Tabla 4 Historia de Usuario 2.....	30
Tabla 5 Historia de Usuario 3.....	31
Tabla 6. Historia de Usuario 4.....	31
Tabla 7. Historia de Usuario 5.....	33
Tabla 8. Historia de Usuario 6.....	34
Tabla 9 Historia de Usuario 7.....	35
Tabla 10 Historia de Usuario 8.....	36
Tabla 11. Historia de Usuario 9.....	37
Tabla 12. Historia de Usuario 10.....	38
Tabla 13. Historia de Usuario 11.....	39
Tabla 14. Historia de Usuario 12.....	39
Tabla 15. Historia de Usuario 13.....	40
Tabla 16. Historia de Usuario 14.....	41
Tabla 17. Historia de Usuario 15.....	41
Tabla 18. Historia de Usuario 16.....	42
Tabla 19. Historia de Usuario 17.....	43
Tabla 20. Historia de Usuario 18.....	43
Tabla 21. Historia de Usuario 19.....	44
Tabla 22. Historia de Usuario 20.....	45
Tabla 23. Historia de Usuario 21.....	45
Tabla 24. Historia de Usuario 22.....	46
Tabla 25. Historia de Usuario 23.....	48
Tabla 26. Historia de Usuario 24.....	50
Tabla 27. Product Backlog.....	51
Tabla 28. Tabla Socios.....	58
Tabla 29. Tabla Fallecidos.....	59
Tabla 30. Tabla Bloques.....	59
Tabla 31. Tabla Nichos.....	60
Tabla 32. Tabla Fallecido_Nicho.....	60
Tabla 33. Tabla Pagos.....	61
Tabla 34. Usuarios.....	61
Tabla 35. Sprint Backlog – Sprint 1.....	62
Tabla 36 Revisión - Criterios de Aceptación Sprint 1.....	70
Tabla 37. Sprint Backlog – Sprint 2.....	76
Tabla 38 Revisión - Criterios de Aceptación Sprint 2.....	83
Tabla 39. Sprint Backlog – Sprint 3.....	87
Tabla 40 Revisión - Criterios de Aceptación Sprint 3.....	103
Tabla 41. Sprint Backlog – Sprint 4.....	107
Tabla 42 Revisión - Criterios de Aceptación Sprint 4.....	114
Tabla 43 Encuesta con el Modelo DeLone & McLean aplicada en la evaluación de la aplicación web.....	122

<i>Tabla 44. Rangos de valoración según la Media.</i> .....	125
<i>Tabla 45. Resultados de la encuesta [Elaboración Propia]</i> .....	126
Tabla 46 Resultados – Evaluación por dimensión [Elaboración Propia] .....	126
Tabla 47 Porcentaje de la Calidad de la Información. [Elaboración propia]. .....	130
Tabla 48 Porcentaje - Calidad de la Información [Elaboración propia].....	132
Tabla 49 Porcentaje - Calidad de Servicio. [Elaboración propia] .....	134
Tabla 50 Porcentaje - Uso - Intención de Uso. [Elaboración propia].....	136
Tabla 51 Porcentaje - Satisfacción de Usuario. [Elaboración propia].....	138
Tabla 52 Porcentaje - Beneficios Netos. [Elaboración propia] .....	141
Tabla 53 Análisis por dimensión del modelo DeLone y McLean. [Elaboración propia] .....	142

## RESUMEN

Este proyecto de titulación explica el desarrollo e implementación de una aplicación web enfocada a optimizar la gestión administrativa del Cementerio Indígena de Otavalo, dirigido por la directiva de la organización UNORICO SAMASHUNCHIK. Para su elaboración, se aplicó la metodología ágil SCRUM y el patrón de diseño Modelo-Vista-Controlador (MVC), usando el Framework Laravel y administrador de base de datos PostgreSQL. Al inicio se desarrolló en un ambiente local, la aplicación termino con un cambio exitoso a la infraestructura en la nube de Supabase en la fase final, asegurando seguridad, accesibilidad y flexibilidad de los datos. La herramienta digitaliza procesos claves como registro, control de socios, fallecidos y asignaciones.

La evaluación de éxito se basó en el modelo de DeLone y McLean. Empleando estadística descriptiva usada en una muestra censal de cuatro usuarios, los resultados revelaron una aprobación excepcional con 0% de desaprobación general. Las dimensiones como los Beneficios Netos y Satisfacción del Usuario obtuvieron porcentajes altos como el 95% y 91.66% verificaron que la aplicación tiene un funcionamiento óptimo, resaltando como influyo en la transparencia y protección de perdida de información. Se determina que la aplicación web es una solución tecnológica totalmente factible que cumple totalmente las necesidades la organización.

Palabras clave: Aplicación web, gestión administrativa, Laravel, modelo DeLone y McLean, Supabase, MVC, digitalización de procesos.

## **ABSTRACT**

This degree project explains the development and implementation of a web application focused on optimizing the administrative management of the Otavalo Indigenous Cemetery, run by the board of directors of the UNORICO SAMASHUNCHIK organization. The agile SCRUM methodology and the Model-View-Controller (MVC) design pattern were applied to its development, using the Laravel Framework and PostgreSQL database manager. Initially developed in a local environment, the application was successfully migrated to the Supabase cloud infrastructure in the final phase, ensuring data security, accessibility, and flexibility. The tool digitizes key processes such as registration, member control, deceased members, and assignments.

The success evaluation was based on the DeLone and McLean model. Using descriptive statistics on a census sample of four users, the results revealed exceptional approval with 0% overall disapproval. Dimensions such as Net Benefits and User Satisfaction obtained high percentages of 95% and 91.66%, verifying that the application functions optimally, highlighting its influence on transparency and protection against information loss. It is determined that the web application is a fully feasible technological solution that fully meets the organization's needs.

Keywords: Web application, administrative management, Laravel, DeLone and McLean model, Supabase, MVC, process digitization.

# INTRODUCCIÓN

## 1.1 Tema

Desarrollo de una aplicación web para la gestión del cementerio Indígena de Otavalo de la Organización UNORICO SAMASHUNCHIK utilizando el Framework Laravel y aplicando el marco de trabajo Scrum

## 1.2 Planteamiento del problema

La ausencia de automatización en la gestión de cementerios produce diferentes problemáticas notables. Fundamentalmente, se aprecia una incapacidad en el manejo y acceso a la información, ya que los registros manuales dificultan las búsquedas [1], aumentan el papeleo y complican el control y la administración de datos importantes como: pagos, traslados, exhumaciones, entre otros [2]. Esta situación dificulta la interacción fluida y efectiva con los usuarios y las entidades, afectando la calidad del servicio brindado [3]. La dependencia de archivos físicos involucra el peligro de pérdida, deterioro e información duplicada, también complica las actualizaciones y la compilación de datos para ver el estado del cementerio [3].

Además, la falta de aplicaciones digitales limita la organización y la gestión del espacio, produciendo desorden y conflictos entre usuarios por la ocupación ilegal de la tierra [4]. El manejo manual de documentos conlleva a un control inapropiado de los pagos, perjudicando los ingresos de la organización [1].

Actualmente, el cementerio indígena de Otavalo, gestionado por la Asociación UNORICO SAMASUNCHIK realiza su administración mediante registros físicos, el cual ha provocado pérdida de información y degradación. Este manejo de la información causa una alta carga administrativa para el personal que atiende a más de 8.745 habitantes pertenecientes a 68 comunidades indígenas, dificultando el control de socios, nichos, fallecidos, entre otros. Impidiendo la transparencia y eficacia en los procesos que se llevan a cabo.

A continuación, se presenta el árbol de problemas desarrollado.

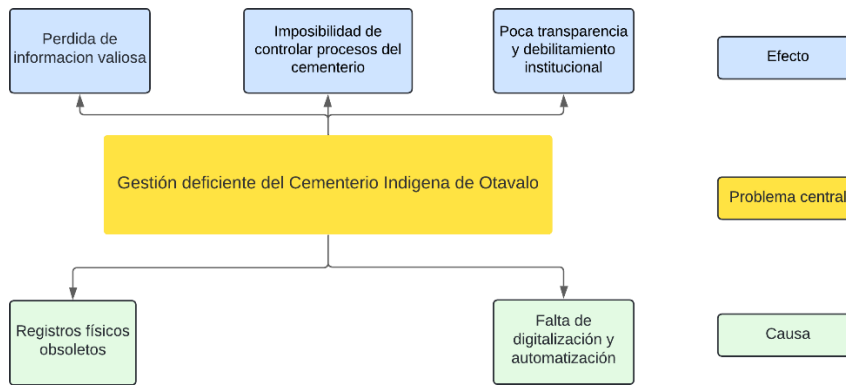


Figura 1. Árbol de problemas

Nota. Elaboración Propia

### 1.3 Objetivos

#### 1.3.1 Objetivo General

Desarrollar una aplicación web para la gestión del cementerio Indígena de Otavalo de la Organización UNORICO SAMASHUNCHIK utilizando el Framework Laravel y aplicando el marco de trabajo Scrum.

#### 1.3.2 Objetivos Específicos

- Elaborar un marco conceptual y técnico para el estudio de desarrollo de aplicaciones web aplicados a la gestión de cementerios, con perspectiva en necesidades culturales y administrativas.
- Desarrollar una aplicación web para mejorar la gestión del cementerio utilizando Laravel y aplicando la metodología de desarrollo Scrum.
- Validar la aplicación haciendo uso del Modelo de Éxito de Sistemas de Información de DeLone y McLean.

### 1.4 Alcance

El proyecto de trabajo de grado es desarrollar una aplicación web para la gestión del Cementerio Indígena de Otavalo, administrado por la organización UNORICO SAMASHUNCHIK, con el propósito de brindar una solución eficiente y accesible para mejorar los procesos administrativos por medio de la digitalización de los registros y la automatización de actividades importantes.

Para el desarrollo de la aplicación web se usará el Framework Laravel, basado en el lenguaje PHP, renombrado por su robustez, seguridad y la facilidad de mantenimiento [5]. Para el almacenamiento y gestión de datos se empleará PostgreSQL como gestión de base de datos relacional el cual permite una administración segura, confiable y escalable

[6]. El desarrollo será realizado siguiendo la arquitectura basada en Laravel el MVC (Modelo Vista Controlador) que permite una mejor organización del código, escalabilidad y mantenibilidad [7].

Para la validación de la aplicación se implementará el Modelo de DeLone y McLean explicado como un marco teórico sólido y completo el cual busca medir la calidad y uso de los usuarios a través de una encuesta en escala de Likert [8].

La aplicación contará con los siguientes módulos, cada uno con funciones específicas para asegurar la administración eficiente del cementerio:

- **Módulo de Gestión de Usuarios:** Se administrará a los empleados y personal administrativo del cementerio, controlando el acceso mediante roles y permisos.
- **Módulo de Gestión de Socios:** El administrador o personal administrativo que cuente con un rol y permisos necesarios podrá registrar, editar, eliminar y gestionar a los socios.
- **Módulo de Gestión de Fallecidos:** El administrador o personal administrativo que posea un rol y permisos necesarios podrá registrar, editar, eliminar, asignar tumba, asignar socio y gestionar a los fallecidos.
- **Módulo de Gestión de Tumbas y Bloques:** El administrador o personal administrativo que tenga un rol y permisos necesarios será capaz de registrar, editar y controlar la disponibilidad de las tumbas y distribución por bloques.
- **Módulo de Servicios:** El administrador o personal administrativo que posea un rol y permisos necesarios será capaz de registrar, editar y controlar la disponibilidad de los servicios que ofrece el cementerio como: alquiler de carpas, excavación de tumbas, mantenimiento, entre otras.
- **Módulo Facturación Servicios:** El administrador o personal administrativo que posea un rol y permisos necesarios podrá registrar y generar las facturas para los servicios brindados por parte del cementerio.

La arquitectura determinará una conexión directa con la base de datos PostgreSQL como se muestra en la Figura 2.

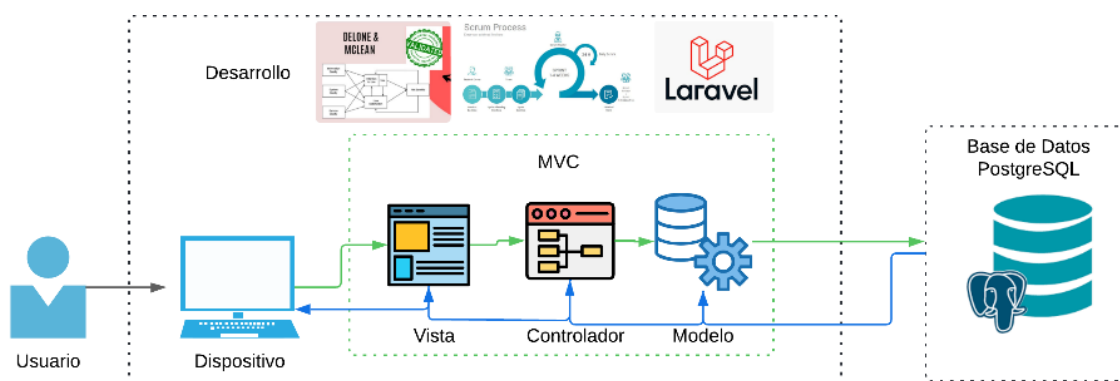


Figura 2. Arquitectura a desarrollar

Nota. Elaboración Propia

## 1.5 Metodología

El presente trabajo de investigación es de tipo aplicada, dado que busca transformar la gestión física y manual mediante el desarrollo de una aplicación web que automatice y mejore la administración del Cementerio Indígena de Otavalo, para el beneficio de 68 comunidades indígenas.

El enfoque metodológico se ajusta a los objetivos del proyecto:

1. Revisión bibliográfica: Se realizará una investigación documental con el propósito de recolectar información acerca de la gestión administrativa de cementerios y el desarrollo de aplicaciones similares. Se hará uso de los repositorios digitales la Universidad Técnica del Norte (UTN), al igual que en otras bases de datos académicas a nivel nacional e internacional. Para mejorar la eficiencia de esta actividad, se usará el gestor bibliográfico Mendeley el cual permite organizar y referenciar las fuentes investigadas [9].
2. Desarrollo de la aplicación: Para alcanzar el segundo objetivo, se aplicará la metodología ágil SCRUM, mediante sus 3 fases principales. En la primera fase: Pre-juego, donde se llevará a cabo el levantamiento de requisitos. Segunda fase: Juego, se realizará el desarrollo de la aplicación web, mediante una serie de iteraciones para brindar mejoras y asegurar el correcto funcionamiento. Tercera fase: Post-juego, se harán pruebas de aceptación y entrega [10].
3. Validación de la aplicación: Con respecto al tercer objetivo, se llevará a cabo la validación de la aplicación desarrollada mediante el Modelo de Éxito de Sistemas de Información DeLone y McLean utilizando una encuesta de

preguntas estructuradas a la escala de Likert para determinar la calidad y uso de los usuarios finales [8].

## **1.6 Justificación**

El presente trabajo se alinea con el Objetivo de Desarrollo Sostenible (ODS) número nueve: el mismo que promueve la construcción de infraestructura, industrialización e innovación [11]. Esta aplicación aporta de forma directa el desarrollo de capacidades tecnológicas en organizaciones comunitarias indígenas, impulsando la digitalización de procesos importantes dentro de una asociación la cual representa a 68 comunidades del cantón Otavalo. Igualmente, este trabajo está vinculado con el objetivo 16 del Plan Nacional de Desarrollo “Creación de Oportunidades 2021-2025”, el cual busca mejorar las capacidades institucionales para una gestión pública de calidad, transparente y moderna [12].

En este ámbito, la política 16.3 determina como importante la implementación de tecnologías digitales en procesos administrativos públicos, enfocada a mejorar la calidad de los servicios, reforzar la descentralización administrativa [12]. Mediante este enfoque, se busca comprender y resolver limitaciones estructurales y operativas que produce la gestión manual de la información en sitios comunitarios. Presentando una alternativa tecnológica que asegure la transparencia, trazabilidad y preservación documental a largo plazo.

### **Beneficiarios**

- **Beneficiarios directos:**
  - Organización UNORICO SAMASHUNCHIK
  - Personal administrativo
- **Beneficiarios indirectos:**
  - 8745 socios del cementerio
  - 68 comunidades indígenas

### **Justificación Tecnológica**

El desarrollo de la aplicación web para la gestión del cementerio indígena de Otavalo es esencialmente necesario la incorporación de herramientas digitales para sustituir los procesos manuales que están propensos a errores, pérdidas y duplicidad de información. La incorporación de tecnologías digitales como Laravel permitirá desarrollar una plataforma robusta, adaptable y de sencillo mantenimiento, reforzando la eficacia y trazabilidad de los datos [5]. Esta propuesta se enmarca en la “Política de

Transformación Digital del Ecuador 2025-2030“, que busca promover la utilización de herramientas tecnológicas y la digitalización de servicios públicos para fortalecer la eficiencia administrativa y calidad de vida de los pueblos indígenas [13].

### Justificación Social

La implementación de la aplicación web para la gestión del Cementerio se adapta con el “Objetivo de Desarrollo Sostenible” número 9, el cual busca construir infraestructuras resilientes, industrialización sostenible y promover la innovación [11]. El objetivo en cuestión reconoce que, al digitalizar los procesos administrativos, se incrementa la eficacia y promueve la inclusión social, mejorando la accesibilidad a servicios fundamentales para las comunidades indígenas.

### 1.7 Antecedentes

*Tabla 1. Antecedentes*

Investigación	Aporte
<b>Contexto: Local</b> Desarrollo de un sistema web para registro, control y administración de bienes inmuebles[4].	Este trabajo se enfoca en la gestión de bienes inmuebles mediante el uso de herramientas como Laravel, PostGIS. Mientras que mi propuesta se basa en el uso de las mismas herramientas, pero para la administración del cementerio Indígena de Otavalo.
<b>Contexto: Local</b> Desarrollo de una aplicación web para mejorar el control y gestión de procesos de la empresa textil Acuatex[8].	En la tesis presentada se empleó el Modelo de Éxito y Sistemas de la Información de DeLone y McLean para la validación del desarrollo lo que permite medir el nivel de éxito con las diferentes dimensiones de evaluación. El estudio puede ser usado como guía referencial para determinar el proceso de validación de la aplicación web de gestión del cementerio.
<b>Contexto: Nacional</b> Implementación de un sistema de inventario informático mediante Generación de Código QR [14].	Este trabajo hace el uso de la tecnología de código QR para la identificación de bienes a través del escaneo mediante un dispositivo móvil. Mientras, mi tesis toma la propuesta como referencia para la gestión del cementerio el cual generará a cada nicho un código QR para la visualización de información específica.

<p><b>Contexto: Nacional</b> Implementación de un sistema web para la gestión del cementerio de Guano haciendo uso del Framework CodeIgniter[15].</p>	<p>En la tesis presentada se hace uso del Framework CodeIgniter que usa el patrón MVC (Modelo, Vista, Controlador) y la evaluación del rendimiento mediante el modelo FURPS. Mi aporte se basa en la implementación del Framework Laravel, PostgreSQL y para la validación de la aplicación se utilizará del modelo de DeLone y McLean mediante una encuesta con escala de Likert para conocer la calidad y uso por los usuarios.</p>
<p><b>Contexto: Internacional</b> Aplicación web para la administración de los procesos del cementerio municipal de Granada[16].</p>	<p>En el trabajo de grado desarrolló un sistema para la administración del cementerio mediante el uso de herramientas como: ASP.NET, Microsoft SQL Server, Power BI. El presente trabajo tiene como aporte el uso de herramientas libres siendo estas Laravel, PostgreSQL las cuales son disponibles y permiten una solución en costos para el cementerio.</p>

# CAPÍTULO I - MARCO TEÓRICO

## 2.1 Gestión de Cementerios

### 2.1.1 Procesos de administración

Los procesos de administración en los cementerios son acciones organizadas y procesos planificados para asegurar la gestión eficiente y la conformidad en brindar servicios funerarios y de sepultura a las personas [17].

La administración ha ido avanzando de manera considerable, cambiando de un control manual y a veces religioso a aplicaciones más completas y administrativas. Antiguamente en sitios como el Consejo Municipal de Lusaka en el país de Zambia, este proceso era básicamente manual, lo cual causaba problemas considerables como dificultad para actualizar los registros, localizar las tumbas de sepultura, localizar cementerios y brindar un ingreso público apropiado a la información [18]. En diversas culturas, la ausencia de documentación oficial de los lugares de entierro ha causado impacto en procesos tal como identificación de tumbas, falta de terrenos apropiados, presupuesto limitado para el mantenimiento [18].

También se debe de tomar en cuenta que una inadecuada gestión de registros, principalmente en cementerios antiguos y descuidados, suele contener información desactualizada, incompleta o perdida, provocando complicaciones a las familias cuando buscan la ubicación de tumbas [18]. Para afrontar estas situaciones difíciles, la gestión municipal ha incorporado puntos de vista más estructurados y lógicos, siendo controlados principalmente por autoridades municipales locales [19].

Las gestiones que se realizan buscan soluciones estables, tales como: promover la cremación, permitir diversos entierros en la misma porción de tierra, dar uso nuevamente a cementerios antiguos, para mejorar el manejo de tierras y equilibrar presupuesto, por el motivo de que los entierros tradicionales suelen ser apoyados y los ingresos no son suficientes para los gastos operativos [20].

### 2.1.2 Problemas en la gestión manual

La gestión manual en los cementerios, tal como se describe en las fuentes, implica problemas relevantes que impactan la eficiencia y la calidad del servicio. Primeramente, los registros y el control de documentos que se realizan manualmente causan que la búsqueda de información sea lenta y el manejo de los datos se dificulte [1]. Este método es considerado completamente anticuado y gasta una cantidad excesiva de tiempo para la

gestión administrativa, lo que aumenta la carga laboral para el personal [1]. Con el constante incremento de la cantidad de registros y de fallecidos, se convierte constantemente más difícil hallar la información necesaria lo antes posible [21]. Igualmente, las consultas para las familias y visitantes acaban siendo incómodas y requieren demasiado tiempo, ya que el personal debe recorrer físicamente el cementerio tumba por tumba para comprobar la disponibilidad de espacios o encontrar información precisa [2].

Segundo, la dependencia de archivos físicos y aplicaciones de oficina para la administración implica serios problemas de veracidad y disponibilidad de datos [1]. Existe peligro de pérdida de datos por la degradación de los archivos físicos y un impedimento para controlar la cantidad y variedad de información [1]. A menudo, la información no se encuentra correctamente organizada, clasificada o completa, y se presenta duplicación de los oficios o contratos para un mismo nicho o tumba [2]. El sistema manual, en varios casos, no guarda copias de seguridad con frecuencia solo una persona puede acceder y controlar la información, lo que lo hace ineficiente y vulnerable a errores graves [1]. Esta gestión práctica y anticuada de los espacios fúnebres también dificulta tener información adecuada para la planificación a largo plazo del cementerio [2].

### **2.1.3 Beneficios de la digitalización en cementerios**

La digitalización en los cementerios, como se indica en los casos del Cementerio “José María Azael Franco Guerrero” y el Cementerio Municipal de Cuyabeno, brinda beneficios importantes en la gestión de la información. Permite el almacenamiento, administración y organización de datos desde una base de datos fuerte, lo que reduce los tiempos de consulta y resuelve dificultades relacionadas con el manejo manual de la información, como el uso de hojas de cálculo de Excel [22]. Esto garantiza una información actualizada y precisa, facilitando un control eficiente del inventario de nichos, tumbas y espacios, también de disminuir el tiempo en procesos como el alquiler o la compra de estos.

Además de la mejora administrativa, la integración de sistemas de información geográfica (SIG) en los cementerios simplifica intensamente la búsqueda y ubicación de nichos, tumbas, servicios funerarios, infraestructura para usuarios y visitantes [22]. Estos sistemas permiten a las personas encontrar la ubicación exacta de sus seres queridos o puntos de interés en poco tiempo, habitualmente mediante mapas interactivos y georreferenciados que pueden indicar rutas y detalles visuales como fotos o videos [22].

Esta organización no solo mejora el tiempo y previene la información duplicada, sino que también mejora la experiencia general del usuario y la disponibilidad de la información [23].

#### **2.1.4 Aplicaciones web para la gestión de cementerios**

Debido a las dificultades presentadas, se ha presentado un aumento progresivo hacia la transformación digital y adopción de aplicaciones web y tecnologías de información geográfica en la administración de cementerios [24]. Las herramientas como GIS (Sistema de Información Geográfica) y GPS (Sistema de Posicionamiento Global) son fundamentales porque nos facilitan la creación de bases de datos geográficas que nos permiten conectar la información de los registros del difunto con su ubicación espacial de la tumba [24]. Gracias a estas tecnologías se nos simplifica considerablemente la consulta, localización, actualización y análisis de datos de cementerios [24].

Mediante esto se logra reconocer los nichos ocupados y disponibles, también se analizarían datos demográficos y genealógicos y organizar una planificación estratégica a futuro de forma más eficiente [24]. La integración de estas tecnologías puede potenciar considerablemente el análisis, precisión de datos y accesibilidad para los encargados y el público en general [24].

Un ejemplo es el mapa en línea de la municipalidad metropolitana de Ankara, se indica cómo las herramientas ayudan a la reducción de carga de trabajo al personal del cementerio y mejorando la búsqueda de los fallecidos en especial los días que hay festividades [25]. Hoy en día existen muchas soluciones locales, pero lo óptimo es avanzar a servicios de mapeo que se encuentren en la web, así como el ANC Explorer perteneciente al Cementerio Nacional de Arlington, para alojar y guardar datos en internet, habilitando así las visitas virtuales e indagando a nivel global [24]. Asimismo, existen tecnologías avanzadas como LiDAR, vehículos no tripulados (UAV) y radar de penetración terrestre (GPR), las cuales se están analizando para conseguir datos topográficos sumamente precisos y validar la disponibilidad de lotes, además, reconociendo tumbas que pudieron haber sido identificadas de manera incorrecta o puede que estén perdidas [24].

## **2.2 Tecnologías de desarrollo**

### **2.2.1 Lenguaje de programación PHP**

Fue implementado en noviembre de 1997, desde entonces PHP ha evolucionado considerablemente, incluyendo características esenciales como la Programación

Orientada a Objetos conocida por sus siglas POO a partir del PHP 3 y perfeccionada en PHP 4, creció su popularidad, consiste en la capacidad para desarrollar aplicaciones web dinámicas [26].

El lenguaje PHP, cuyas siglas significan Hypertext Pre-Processor, se entiende como un lenguaje de "scripting" de propósito general y código abierto diseñado principalmente para el desarrollo web; su capacidad para ser embebido directamente en páginas HTML lo hace ágil y potente [27]. La sintaxis de PHP se basa en lenguajes como Java, C y Perl, lo que facilita su aprendizaje. El objetivo fundamental de PHP es permitir a los programadores web escribir rápidamente páginas web dinámicas [27]. A diferencia de las tecnologías del lado del cliente como JavaScript, la programación en PHP se ejecuta en el servidor, generando HTML que luego es enviado al cliente, lo que impide que el usuario acceda o modifique el código subyacente [14]. Incluso se puede configurar el servidor web para procesar la totalidad de los ficheros HTML con PHP.

Entre las características y ventajas clave de PHP se destaca que, al ser un lenguaje libre, cuenta con una licencia abierta y una gran comunidad de soporte en línea [27]. Es multiplataforma, lo que garantiza una gran compatibilidad y adaptabilidad con diversos sistemas operativos [27]. Su trabajo del lado del servidor les confiere compatibilidad y fácil adaptación a distintos tipos de bases de datos, como MySQL. La escalabilidad es otra ventaja importante, ya que Frameworks como Laravel permiten generar proyectos modulares y de fácil ampliación [27]. Además de su facilidad de aprendizaje, posee un alto rendimiento, cuenta con interfaces para una gran cantidad de sistemas de bases de datos, bibliotecas incorporadas para tareas web comunes, bajo costo, portabilidad y accesibilidad a su código abierto [27]. En cuanto a las versiones, se menciona la utilización de PHP 7 para el backend, con requisitos específicos como PHP 7.1 o PHP >= 8.0 para la implementación en el servidor, y se ha usado PHP 7.2.19 en proyectos [14].

Los Frameworks que trabajan con PHP y son muy recomendados son Laravel y CodeIgniter para el desarrollo de aplicaciones de escala media a alta, por motivo de sus varias características incorporadas, seguridad mejorada y la facilidad de cooperación entre programadores en un ecosistema estructurado [26]. Acerca de Laravel se diferencia por su estabilidad, actualizaciones que recibe y la estructura que dispone para proyectos empresariales, a la vez que CodeIgniter brinda un balance entre características y ligereza siendo perfecto para proyectos pequeños a medianos [26].

## 2.2.2 Arquitectura MVC

El patrón de diseño Modelo Vista Controlador es una arquitectura esencial en la ingeniería de software, sobre todo en aplicaciones con la interfaz de usuario [28]. Su innovación se acredita a Trygve Reenskaug en los años 1970; sin embargo, fue definido técnicamente por Glenn E. Krasner, Stephen T. Pope y ParcPlace Systems en el año 1988 [28], [29]. Laravel es un framework que usa PHP, el cual utiliza este patrón para desarrollar aplicaciones web [7].

El objetivo principal del MVC es la división de preocupaciones [29]. Esto quiere decir que el código de la aplicación se separa en tres capas diferentes como lógica de negocios (Modelos), control (controladores) y la demostración de datos (Vistas) [7]. Esta división mejora aspectos del código como la mantenibilidad, legibilidad y ayuda la colaboración entre desarrolladores tanto de frontend como de backend [30].

- La capa del Modelo se encarga de la comunicación con la base de datos. Laravel incluye un componente opcional llamado Eloquent, una clase que se hereda a los modelos para facilitar la conexión con los motores de bases de datos [4].
- La capa de la Vista es donde se crean las páginas web, utilizando un motor de plantillas denominado Blade. Blade tiene como objetivo proporcionar un código limpio y fácil de comprender para las vistas, y los archivos de plantilla se crean con la extensión blade.php [4].
- La capa del Controlador contiene la lógica de negocio de la aplicación. Sus funcionalidades incluyen la recuperación de entradas de la base de datos para listarlas, actualizar, eliminar o buscar información, añadir nuevos registros y generar mensajes de confirmación de las funciones básicas [4]. Se recomienda el uso de Laravel en proyectos a gran escala debido a su complejidad en características, su capacidad para estandarizar el proceso de desarrollo y su gran escalabilidad, lo que puede mejorar la eficiencia [4].

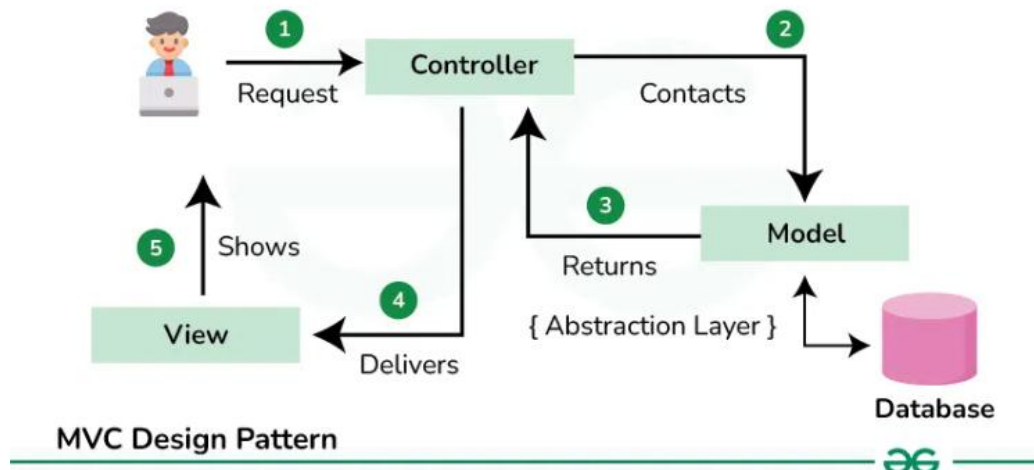


Figura 3. Patrón de diseño MVC.

Fuente: [31]

### 2.2.3 Framework Laravel

Laravel es un Framework de código libre para desarrollar aplicaciones web haciendo uso del lenguaje PHP, tiene una sintaxis clara y elegante, lo que pretende facilitar el desarrollo al reducir tareas habituales en gran parte de proyectos web, como: autenticación, enrutamiento, sesiones y almacenamiento caché [32]. Fue presentada por primera vez en el 9 de junio del 2011, fue desarrollada por Taylor Otwell para sustituir la ausencia de funciones principales, así como en la autenticación de usuarios, pese a que al inicio no se incluía el patrón MVC se lo implementó en la segunda actualización luego de seis meses [30].

En el año 2015 destacó por ser uno de los Frameworks PHP más destacados [29]. Laravel brinda muchos beneficios y ventajas, tales como la integración de varias librerías fuertes y herramientas útiles para los desarrolladores [7]. De igual manera puede servir como backend de API para una aplicación de JavaScript de una única página o de una aplicación móvil; esto se puede utilizar para la autenticación y almacenamiento /recuperación de datos para la aplicación [33].

Laravel es un Framework de código abierto para desarrollar en lenguaje PHP [4]. Su filosofía se enfoca en lograr un código que sea expresivo y elegante, y está diseñado específicamente para la creación de aplicaciones y servicios web [4]. Se destaca por haber

sido concebido para mejorar la calidad del software y reducir tanto los costos de desarrollo como los de mantenimiento [4]. Además, busca mejorar la experiencia en el desarrollo al ofrecer una sintaxis clara y expresiva, y un conjunto básico de funcionalidades que permiten ahorrar horas en la implementación de proyectos [4].

En cuanto a sus características, Laravel integra numerosas funciones de seguridad, incluyendo la gestión de contraseñas, cifrado y validación de datos. Utiliza seguridad OpenSSL y cifrado AES-256-CBC para la encriptación [4]. Se ha comprobado que es beneficioso para la creación de sitios web de comercio electrónico y puede mejorar la reputación de los sitios web [4]. Es un framework modular y extensible, lo que significa que cuenta con un directorio llamado Packalyst que contiene más de 5.500 paquetes para añadir cualquier funcionalidad adicional que se desee [4]. También permite crear el inicio de sesión con comandos y ofrece autenticación de usuarios de forma nativa [4]. Laravel es compatible con múltiples motores de base de datos como MySQL, PostgreSQL, MSSQL y SQLite, lo que asegura que no presenta problemas de compatibilidad [4].

La arquitectura de Laravel adopta el patrón tradicional MVC (Modelo-Vista-Controlador). Este patrón separa la lógica de negocios, la lógica de entradas y la presentación, facilitando la identificación de errores en el código al tenerlo separado [29].

Para hacer más fácil el desarrollo, este framework hace uso de Composer como su administrador de paquetes los cuales permiten administrar y declarar las librerías que sean necesarias [29]. De igual manera cuenta con Artisan, el cual es una interfaz de línea de comandos incorporada que optimiza las tareas repetitivas, así como la creación de código esqueleto, es decir, un código básico, manejo de esquemas de bases de datos a través de archivos de migración y la generación del controlador, modelo, middleware [29].

#### **2.2.4 Base de datos PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos de objetos relacionales, de código abierto, con más de 35 años de desarrollo y una reputación sólida de arquitectura robusta, fiabilidad, integridad de datos y robustez [34]. Originalmente conocido como Postgres, se inició como parte del proyecto POSTGRES de la Universidad de California en Berkeley en 1986, bajo la dirección del profesor Michael Stonebraker [35]. En 1995,

se cambió el nombre a Postgres95 y en 1996, el proyecto se renombró a PostgreSQL para reflejar la relación entre el sistema original de Postgres y las versiones más recientes con capacidad SQL [35].

PostgreSQL está diseñado para ayudar a los desarrolladores a crear aplicaciones y a los administradores a proteger la integridad de los datos [34]. Permite almacenar y escalar de forma segura cargas de trabajo de datos más complejas [34]. El desarrollo de PostgreSQL se ha centrado en ampliar sus funciones y capacidades, incluyendo una amplia gama de tipos de datos, sistemas de integridad de datos, concurrencia, fiabilidad, seguridad, extensibilidad y herramientas de internacionalización [34]. Es una herramienta muy versátil, adecuada para proyectos sin estadísticas, y se utiliza para organizar datos estructurados, semiestructurados y polimórficos, así como para manejar datos espaciales y de geometría [34].

## **2.3 Metodología SCRUM**

### **2.3.1 Principios de la Metodología**

Scrum es un marco de trabajo ágil; este se centra en realizar actividades de manera iterativa y con mejoras constantes [36]. Se levantó sobre las bases del empirismo, el cual asegura que el conocimiento procede de la experiencia y las decisiones son tomadas de acuerdo con lo observado [36]. Scrum impulsa la colaboración entre equipos para obtener un trabajo de gran impacto y reduce la complejidad en el desarrollo de productos para responder a las expectativas del cliente [37].

Scrum lleva a cabo su enfoque iterativo e incremental mediante tres pilares empíricos importantes: Transparencia, Inspección y Adaptación [37]. Estos pilares posibilitan a los equipos Scrum mejorar la previsibilidad y reducir riesgos para que funcionen de manera eficaz al incorporar cuatro eventos formales para la inspección y adaptación, lo que se denomina Sprint [37].

- **Transparencia:** Conlleva que el proceso y el trabajo tienen que ser visibles para todos los participantes, para todos los que desarrollan el trabajo como para aquellos que lo reciben [36].

- Las decisiones importantes se fundamentan en el estado observado de sus tres artefactos formales (Product Backlog, Sprint Backlog e Incremento), dado que la ambigüedad en la evaluación puede inducir a errores y daños innecesarios [37].
- Inspección: Se relaciona con la necesidad de evaluar los artefactos de Scrum y el avance para los objetivos establecidos de forma regular y responsable. Esto facilita identificar variaciones o problemas inesperados que no se consideraron en la planificación y permite una fácil adaptación [36].
- Adaptación: Quiere decir que, si algún aspecto del proceso se aleja de lo establecido o el resultado es insatisfactorio, se deben realizar ajustes inmediatos para minimizar desviaciones y mejorar el resultado [37].

### **2.3.2 Etapa 1.-Pre-Juego**

Pre-Juego, conocido como Sprint 0 o etapa de preparación, comienza la fase inicial del proyecto. En esta fase se determina la necesidad o el problema que se va a resolver y se forma el equipo de trabajo, otorgando roles de Scrum (Product Owner, Scrum Master y Developers) a cada integrante y partes interesadas en el proyecto [36].

Roles de Scrum:

- Product Owner: Se encarga de conseguir los requisitos del usuario.
- Scrum Master: Es el líder del equipo, el cual se asegura de que el equipo trabaje de manera eficiente.
- Developers: Son quienes se encargan del desarrollo del software.

Este se dedica a la planificación del proyecto de software, incorporando la definición de la arquitectura, creación de un esquema para la base de datos y el desarrollo de un diagrama de procesos [37]. Es relevante resaltar que, en comparación con los sprints regulares, el Sprint 0 no intenta proporcionar un avance del producto funcional, más bien trata de establecer las bases y la preparación para el proyecto [37].

### **2.3.3 Etapa 2.-Juego**

El juego es la etapa de desarrollo del producto donde se planifica y realiza el trabajo mediante iteraciones conocidas como Sprints [36]. Estos Sprints son periodos de desarrollo que tienen un tiempo límite, que pueden variar entre una y cuatro semanas, de acuerdo con el nivel de dificultad y el riesgo para el proyecto [37].

Este tiene como objetivo entregar software funcional por cada Sprint para impulsar la retroalimentación del cliente, supervisar el proceso de manera más productiva y permitir entregas previstas si se requiere [37].

Durante esta etapa, se inició desde el Product Backlog para desarrollar el Sprint Backlog, el cual es una lista de elementos y tareas seleccionados para el Sprint, conocidos como Historias de Usuarios. El equipo establece el tiempo de duración de cada Sprint, procurando que los tiempos sean equivalentes para una mejor organización [37]. El Scrum Master asigna las tareas a los Developers, los cuales las ejecutan dentro del plazo asignado para el Sprint [36].

En el transcurso del desarrollo del Sprint, el Scrum Master atiende las dudas e inquietudes del equipo a través de reuniones diarias conocidas como Daily Scrum, que tienen una duración de 15 minutos [36]. Estas tienen como objetivo revisar el avance del progreso enfocado al objetivo del Sprint, ajustar el Product Backlog si requiere, detectar problemas y fomentar la toma de decisiones rápidas [36].

Después de finalizar el Sprint, se lleva a cabo un Sprint Review, es una reunión de casi 4 horas para el Sprint del mes con los Scrum Team y los Stakeholders, para analizar las actividades realizadas, comprobar el cumplimiento de los objetivos y adaptar el Product Backlog para el siguiente Sprint, donde los Developers comunican las dificultades halladas y se organiza una mejora continua para mejorar la calidad y efectividad en futuros Sprints [36].

### **2.3.4 Etapa 3.-Post-Juego**

Post-Juego, esta es la etapa final de Scrum, donde se lleva a cabo la entrega del producto y se terminan las actividades del proyecto [37]. El equipo que está liderado por

el Scrum Master y Developers que se juntan para identificar mejoras conseguidas durante los Sprints.

El Scrum Master se responsabiliza de registrar los avances, aquellos que son revisados por el Product Owner en conjunto con el cliente con el fin de garantizar que se hayan realizado los requisitos y objetivos programados [37]. Si los avances no cumplen con las expectativas, el proceso puede volver a la fase Pre-juego para reorganizar los objetivos. Si todo está de acuerdo, se realiza la entrega final del producto [36].

En esta etapa además se incorpora la ejecución de pruebas de aceptación, donde los usuarios o profesionales del negocio confirman las funcionalidades establecidas en la fase inicial (requisitos). También se realiza el despliegue del producto y esto conlleva configuración de servidores, base de datos, seguridad y pruebas finales para asegurar el correcto funcionamiento de la aplicación [37].

## **2.4 Modelo de Éxito y Sistemas de Información de DeLone y McLean**

### **2.4.1 Aspectos fundamentales del Modelo**

El modelo DeLone and McLean fue originario en el año 1992 y actualizado en el 2003; este fue diseñado para lograr medir la eficiencia y el valor que tiene el sistema [38]. El objetivo principal de este fue organizar la investigación que se realizó previamente acerca del éxito de los sistemas, brindando una estructura entendible para tomar en cuenta como “éxito” en la investigación [38].

Se basó en la teoría de la “comunicación” como se transmite la información de manera precisa y la teoría de la “influencia” de la información como esta puede afectar a las personas [38]. La teoría de la comunicación es de parte de Shannon y Weaver del año 1949 y la teoría de influencia de Mason de 1978.

La versión de 1992 propuso seis dimensiones las cuales son: calidad del sistema, calidad de la información, uso, satisfacción del usuario, impacto individual e impacto organizacional [39]. Los autores resaltaron que estas dimensiones no están interconectadas, sino que son interdependientes y para evaluar el éxito de un SI de forma precisa, es fundamental medirlas y/o controlarlas [40]. Omitir esta interdependencia

puede dar lugar a resultados confusos o un análisis incompleto del sistema que se está indagando [38].

En su actualización del 2003 en este modelo se incorporaron nuevas dimensiones, las cuales fueron: Calidad de Servicio y la Intención de Uso, esta última es como una variante de la dimensión de Uso [40]. También consolidó tanto los Impactos Individuales como Organizacionales en una sola y única dimensión nombrada Beneficios Netos [40]. Luego de un tiempo DeLone & McLean propusieron cambiar su nombre de “Beneficios Netos” por “Impactos Netos” para englobar tanto los resultados positivos como los negativos, y aumentaron ciclos de retroalimentación de estos impactos hacia el uso y la satisfacción del usuario, y así también desde el uso y la satisfacción del usuario en dirección a las dimensiones de calidad [40].

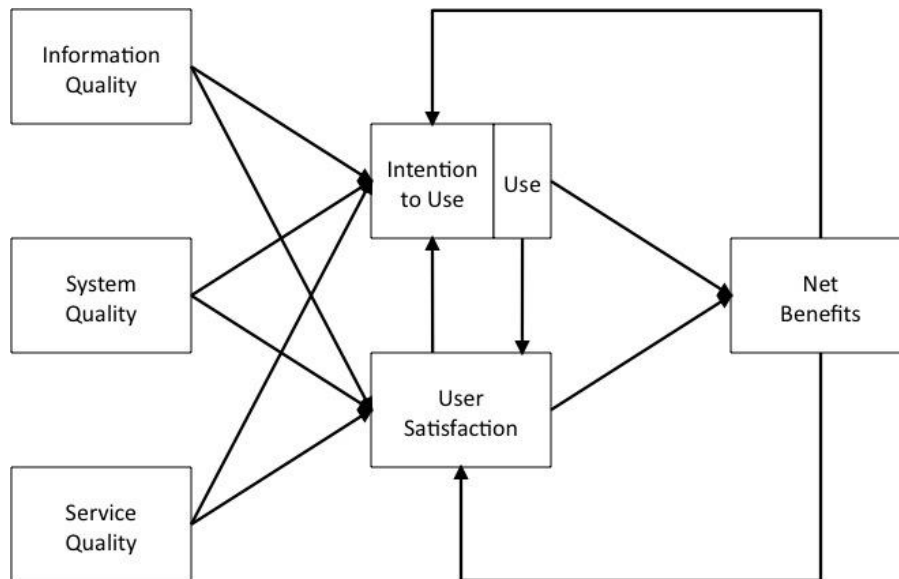
#### **2.4.2 Dimensiones claves para evaluación de aplicaciones: calidad del sistema, uso, satisfacción y beneficios**

Con el propósito de entender el éxito de un SI, es importante comprender la definición y evaluación de sus distintas dimensiones, entre las que resaltan Calidad del Sistema, Uso, Satisfacción del Usuario y Beneficios Netos [38].

- Calidad del Sistema (SQ): En esta dimensión se habla acerca de las características anheladas de un sistema de información como, por ejemplo: facilidad de uso, flexibilidad del sistema, fiabilidad, intuición, velocidad de procesamiento en respuesta y la sofisticación [41], [42]. Se lo evalúa regularmente por cómo funciona el hardware y el software de manera integral [43]. Con esta dimensión se espera que influya de manera positiva considerablemente en el uso del sistema, así como en la satisfacción del usuario [42].
- Uso (SU): Esta dimensión significa el nivel y la manera en que los empleados y los clientes usan las capacidades de un sistema de información; dentro de estas medidas pueden comprender la cantidad de uso, su frecuencia, naturaleza, eficiencia y escalabilidad [40]. Se lo considera una medida de éxito apropiada en su mayoría [38]. Se logra diferenciar si el uso es voluntario y obligatorio, incluso si es de manera obligada; la calidad y frecuencia de uso pueden cambiar los beneficios [38].

- Satisfacción del Usuario (US): En esta dimensión se evalúa el grado de satisfacción que tienen los usuarios con informes mediante sitios web y por medio de los servicios de soporte [40]. Es visto como una de las medidas más fundamentales del éxito del sistema, principalmente cuando el uso es requerido dado que el este por cuenta propia no podría ser un indicador suficiente [41]. La satisfacción registra cómo se siente el usuario a lo largo de la interacción que tiene con el sistema [43].
- Beneficios Netos (NB): Representa la última medida del éxito del SI, estableciendo el alcance en que el sistema aporte o no aporte al éxito de las organizaciones, grupos, individuos, naciones e industrias [38]. Esta dimensión une tanto los “impactos organizacionales” y los “impactos individuales” del que era el modelo de 1992 [43]. Su terminología “Neto” es esencial ya que conlleva la resolución de los resultados positivos y negativos acerca del sistema [38]. Se evalúa mediante el impacto en la toma de decisiones, desarrollo económico, ventas, productividad, mejora en las ganancias y eficiencia del mercado [40].

Las dimensiones mencionadas anteriormente se integran de una forma interdependiente y dinámica internamente en el modelo D&M [38]. Entre las dimensiones de calidad que lo conforman están: calidad del sistema, calidad de la información y calidad del servicio; son antecedentes que afectan el uso y la satisfacción del usuario [38]. Además estas dimensiones anteriores están fuertemente vinculadas: el uso debe brindar satisfacción al usuario durante el proceso, pero si se obtiene una experiencia favorable, se obtendrá una mayor satisfacción [43]. Al final, al igual que el uso, la satisfacción del usuario dirigen a los beneficios netos [38]. Con el propósito de evaluar con precisión el éxito de un SI, el modelo propone que deben de ser tomadas en cuenta todas sus dimensiones y sus relaciones analizadas [38].



*Figura 4. Dimensiones del Modelo DeLone and McLean*

*Fuente: [44]*

### **2.4.3 Métodos de recopilación de datos**

Los métodos de recolección de datos que se usaron en las investigaciones que hacen uso del Modelo de Éxito de los Sistemas de información de DeLone y McLean se enfocan principalmente en enfoques cuantitativos [42]. Generalmente estos estudios se fundamentan en encuestas para adquirir información, y la herramienta clave para recolectar datos por lo general son cuestionarios [41]. Estos usualmente incluyen escalas de Likert para evaluar los puntos de vista y los niveles de satisfacción por parte de los usuarios [42].

La recolección de datos en estas investigaciones se identifica por diseños de investigación transversal, los cuales tratan de realizar un análisis de datos de una población o muestra en un tiempo específico, lo que permite reconocer las relaciones, tendencias, patrones que se tienen en ese único momento [39]. Sin embargo, con el fin de analizar las relaciones mutuas entre las distintas dimensiones que posee este modelo, se ha propuesto la integración de diseños de investigación longitudinales, lo cual trata de realizar la investigación a una población o muestra a lo largo del tiempo, lo que implicaría una mejor comprensión más profunda del comportamiento para analizar los cambios mencionados anteriormente del diseño transversal [39].

Una parte considerable de la investigación se basa en medidas autorreportadas las cuales tratan de la percepción del usuario acerca del sistema esto se recopila mediante las encuestas o cuestionarios, para tener registros sobre la usabilidad, satisfacción y uso del sistema, también se han implementado en algunos estudios las medidas objetivas son aquellas que son datos observables y medibles estos son recopilados mediante registros de uso, logs y registro de rendimiento y se registran como: números de usuarios activos, el tiempo de respuesta, etc. [43].

También se ha tomado en cuenta que las investigaciones que hacen a nivel de análisis organizacional ya han usado medidas objetivas fundamentadas en características de los sitios web, así como la existencia o falta de funciones específicas, número de vistas mensuales o las visitas recurrentes, adquiriendo estos datos de bases públicas [43].

Esta interdependencia en los métodos de recopilación de datos, que se integra tanto en las mediciones autorreportadas como en las objetivas, permite investigar varias facetas del modelo de éxito de los sistemas de información [43].

#### **2.4.4 Interpretación de resultados**

Con el fin de explicar los datos adquiridos de encuestas o cuestionarios acerca del éxito de un sistema de información (SI), el procedimiento conlleva una evaluación minuciosa que va más allá de una simple recolección de respuestas. Se refiere a cómo comprender las evaluaciones y respuestas que dan los usuarios con las dimensiones que se proponen en el modelo si hay satisfacción para el usuario o no [38].

Lo primero que se hace cuando ya se recolectaron los datos mediante encuestas, es la validación del instrumento [45]. Para lograr esto es a través de evaluaciones de validez y fiabilidad de las mediciones, así como la validez convergente el cual trata de verificar si varias medidas son de un mismo concepto están relacionadas entre sí y que también midan lo mismo y den resultados similares, validez discriminante este en cambio es para asegurarse que las medidas se encarguen de medir lo que se supone que deben y no algo que sea diferente, fiabilidad compuesta se encarga de evaluar si los varios ítems que miden una dimensión están midiendo lo mismo de manera estable y por último el Alfa de Cronbach [41].

## **2.5 Trabajos Relacionados**

### **DESIGN, DEVELOPMENT, AND IMPLEMENTATION OF A DIGITAL BURIAL RECORD-KEEPING AND MANAGEMENT SYSTEM IN BANGLADESH**

En febrero del 2025, Rahman et al.[46] Desarrollaron e implementaron un sistema digital de registro y gestión de defunciones en la ciudad de Daca, Bangladesh, con el propósito de mejorar la documentación de fallecimientos. El estudio tuvo lugar en seis cementerios, aplicando investigación exploratoria para entender los procesos mediante observaciones, entrevistas y un software de registro electrónico que fue hecho con Java y .NET Core. Se desarrollo un prototipo y pruebas con los usuarios interesados; esto involucró demostraciones prácticas con los encargados de cementerios ofreciendo retroalimentación. Luego de consultar con los interesados se desarrolló el sistema integrado el cual está destinado a varios usuarios: Aplicación móvil y Sitio web para el público en general, Plataforma de Registro de defunciones para los encargados y un Panel de control para administradores. También se elaboraron mapas digitales utilizando tecnología de drones. Para la evaluación se lo realizó mediante encuestas cuantitativas (SUS y TAM) para analizar la usabilidad y adopción tecnológica con 200 usuarios y 14 encargados; los resultados de las pruebas de calidad en estos seis aspectos mostraron los siguientes resultados: calidad de la información 80%, calidad del sistema 76%, calidad del servicio 111%, uso 85%, satisfacción 86% y beneficio neto 78%. Todos estos resultados de acuerdo con el Modelo de DeLone y McLean dan 84% lo que significa un rendimiento óptimo. Este estudio presentó limitaciones como dependencia de datos cualitativos ya que no disponía de suficientes datos cuantitativos para la evaluación del sistema y esto limita la generalización en otros entornos como cementerios privados. Al igual que hubo otros desafíos sobre la adaptabilidad tecnológica por los niveles de alfabetización informática, infraestructura y recursos.

### **INTEGRATED IDENTITY TAG SYSTEM FOR BIODIVERSITY COLLECTIONS IN INDONESIAN BOTANICAL GARDEN USING QR CODE**

En mayo del 2024, Apriyanto et al.[47] Presentó el desarrollo e implantación de un sistema de etiquetado de identificación para colecciones de plantas en jardines botánicos de Indonesia, mediante el uso de códigos QR. Su finalidad fue enfrentar los problemas de revisión cotidiana de las plantas como: error de identificación, ausencia del

registro histórico de inspecciones e ineficacia en el proceso de informes. Se aplicó la Metodología de Investigación y Desarrollo (I+D) comenzando en la fase de planificación mediante recolección, análisis de información y especificación de requisitos, desarrollo de la arquitectura del sistema, diseño de interfaces para los códigos QR, páginas y por último la evaluación por medio de pruebas del sistema con usuarios de cinco jardines. Como resultados se obtuvo que el nuevo sistema solucionó eficientemente las dificultades de revisión cotidiana, al incorporarse con el sistema de información de colección de plantas, causando un decremento en la probabilidad de errores en los informes, mejoró la eficiencia operativa en procesos de reporte y dio lugar al monitoreo en vivo de las plantas. Al igual que el sistema brinda información fundamental a los pasajeros para enseñar acerca de la flora. Obteniendo una tasa de éxito del 100%. Pero se identificó una deficiencia a pesar de que se obtuviera una tasa de éxito del 100% en las pruebas de la interfaz; esto no quiere decir que el sistema sea de utilidad o de satisfacción para los usuarios por el hecho de tener solo la funcionalidad de indicar información, lo cual propone una evaluación más profunda.

## **COMPARATIVE STUDY: PERFORMANCE OF MVC FRAMEWORKS ON RDBMS**

En febrero del 2024, Rahman et al.[48] Llevaron a cabo un estudio sobre la comparación del rendimiento de los diferentes Frameworks como Laravel, CodeIgniter y Phalcon con el patrón MVC en operaciones CRUD básicas de base de datos relacionales, para determinar la integración más efectiva en el desarrollo de aplicaciones web. Se usó una metodología de prueba de rendimiento, realizando operaciones CRUD como insertar, actualizar, eliminar, con una enorme cantidad de datos de un millón de registros en bases como MySQL y PostgreSQL y calculando el tiempo medio de ejecución por cada diez mil operaciones. El resultado más destacado fue el Framework Phalcon MVC que presentó un mejor rendimiento en bases de datos relacionales, especialmente con PostgreSQL, sobrepasando a CodeIgniter y Laravel en operaciones básicas. Una deficiencia de este estudio fue que solo se centró en operaciones básicas, lo cual no demostraría el rendimiento en aplicaciones más complicadas.

## **SISTEMA WEB EFICIENTE PARA LA GESTIÓN ADMINISTRATIVA EN UNA EMPRESA DE TRANSPORTES PRIVADA**

En junio del 2024, Campos Montero et al.[49] Desarrollaron una aplicación web para mejorar los procesos de mantenimiento, ventas y gestión del personal en una empresa de transportes particular. Para cumplir con este objetivo hicieron uso de la metodología SCRUM, se siguió el patrón Modelo Vista Controlador MVC y el desarrollo de la aplicación fue hecho en el Framework Laravel 10 con el lenguaje de programación PHP 8.2.4 y con una base MySQL 8.0.34. Se hicieron pruebas con Apache JMeter para analizar el rendimiento, con una capacidad de 6400, 1055 y 500 usuarios al mismo tiempo en los diferentes subsistemas. Como resultado se obtuvo la mejora relevante en rendimiento operacional al disminuir costos y tiempos de procesamiento manual; también se asegura la sostenibilidad económica. Una limitación relevante fue que, aunque se trabajara con la metodología SCRUM, hubo dificultades con el cálculo de los tiempos de desarrollo, superando moderadamente el plazo inicial planteado.

### **PERFORMANCE COMPARISON BETWEEN LARAVEL AND EXPRESSJS FRAMEWORK USING APACHE JMETER**

En enero del 2024, Mangapul Siahaan et al.[50] hicieron una comparación acerca del rendimiento de los Frameworks Laravel y Express.js para el desarrollo de RESTful APIs, centrándose en la contestación de la API al ingreso de datos de estudiantes en la base de datos MySQL, con el propósito de brindar ayuda a los desarrolladores para escoger un Framework más apropiado. Para esto se aplicó una metodología de prueba de rendimiento haciendo uso de Apache JMeter en dos máquinas virtuales con Laravel v10.25.2 con PHP y Express.js (Node.js v18.12.0) y con la base de datos de MySQL v8.0.34. Para la ejecución de las pruebas se enviaron solicitudes HTTP GET para lograr ingresar a los registros de estudiantes, fingiendo entre cien y mil usuarios al mismo tiempo y supervisando el tiempo de respuesta. Su resultado más importante fue que Laravel indicó un mayor rendimiento en comparación con Express.js con un tiempo promedio de 1745.7 ms, mientras que Express.js tuvo un tiempo de 10855.1 ms, pero el uso más eficiente en recursos en CPU 20.14%, Memoria 4.21%, esto para Laravel y los siguientes datos de CPU 28.44%, memoria 8.51% para Express.js. Una limitación esencial de este estudio se debe a que los resultados están siendo muy afectados ya que se basó en un escenario de prueba, lo cual señala que esto no podría ser aplicable a todos los tipos de aplicaciones.

## **WEBSITE-BASED VINTAGE CLOTHING SALES INFORMATION SYSTEM USING THE LARAVEL FRAMEWORK**

En marzo del 2025, Darmalaksana et al.[51] Desarrollaron e implementaron un sistema de información de ventas de ropa vintage para Atlas Shop, con el propósito de incrementar la productividad y la competencia en el negocio, optimizando las transacciones y gestión de ventas. Para este propósito se hizo uso de la metodología Rapid Application Development (RAD) y el Framework Laravel con el lenguaje PHP, haciendo uso de Composer, Bootstrap para la parte visual y responsiva y Midtrans para los pagos en línea mediante vía API. La recopilación de datos se lo efectuó mediante técnicas cuantitativas y encuestas y para la verificación del sistema se lo realizó mediante el modelo UTAUT. Como resultado más importante fue que el sistema realizado ofreció una experiencia satisfactoria al usuario, enfatizando la excelente satisfacción con la simplicidad de uso y el potencial de respuesta de la interfaz, lo cual trajo consigo una mejora relevante en la eficiencia operativa y la competencia del negocio. Un inconveniente encontrado fue la exclusión de una variable (BI3) en la validez, lo que implica una probable falencia en la validez de esa característica de la evaluación del comportamiento del usuario.

## **SYSTEM QUALITY AND THE IMPACT OF THE USE OF THE FLO APPLICATION ON FREE ASSOCIATION WITHIN THE COMMUNITY IN PADANG CITY USING THE DELONE AND MCLEAN METHOD**

En abril del 2025, Nailul Hikmi et al.[52] analizaron la calidad de un sistema de la aplicación Flo y el impacto que esta tienen en las interacciones sociales en la ciudad de Padang, con el propósito de investigar como la aplicación aporta a la calidad de pensamiento libre y el efecto que trae en el comportamiento social en el ámbito de la tecnología. Para esto, se hace uso de una metodología de investigación enfoque mixto el cual combina el método cuantitativo basado en encuestas dirigido por el modelo de éxito de sistemas de información de DeLone a McLean y el método cualitativo descriptivo que se basa en la observación directa y entrevistas a setenta encuestados de la sociedad que fueron seleccionados previamente por muestreo. Estos datos fueron recopilados mediante un cuestionario estructurado y luego fueron analizados haciendo uso de técnicas estadísticas. Su resultado fue que la aplicación Flo tuvo una calidad muy buena, teniendo un puntaje promedio del 84% en la evaluación del Modelo de DeLone y McLean. Esto muestra que la calidad de la aplicación afecta notablemente en la satisfacción y

comportamiento del usuario. No obstante, una limitación importante es que, pese a la facilidad de comunicación y conectividad, se encontraron altos niveles de pensamiento libre y un incremento en casos de comportamientos violentos entre jóvenes, lo cual causa preocupación sobre los valores y normas sociales tradicionales y se propone el desarrollar mejoras en el futuro que fomenten una interacción social positiva y se reduzcan los riesgos.

#### **ANALISIS KEBERHASILAN PENGGUNAAN EDUCATION MANAGEMENT INFORMATION SYSTEM (EMIS) 4.0 MENGGUNAKAN METODE DELONE AND MCLEAN**

En enero del 2025, Sofiani et al.[53] analizaron la calidad y el impacto del uso del sistema de Información Education Mangement Information System (EMIS) 4.0 del Ministerio de Asuntos Religiosos, con el propósito de evaluar el éxito de su nuevo sistema centralizado de gestión y recolección de datos para las escuelas. Esta investigación aplicó una metodología a cuantitativa a través de la repartición de encuestas a ciento diecisiete operadores de MTs en Regencia de Ciamis. Haciendo uso del modelo de éxito de sistema de información de DeLone y McLean para poder calcular sus seis dimensiones claves y procesar los datos con SPSS Y SmartsPLS. Lo que se obtuvo como resultado más importante fue que, todas las dimensiones del sistema (calidad del sistema, calidad de la información, calidad del servicio, uso, satisfacción del usuario y beneficios netos) tuvieron una valoración positiva por los usuarios, mostrando una buena calidad general del sistema. Como limitación encontrada fue que “no todas las variables tienen que influir en otras variables” aparte la investigación solo se basó en opiniones de una pequeña fracción de usuarios de EMIS 4.0 lo cual podría afectar la generalización de los resultados.

## CAPÍTULO II - DESARROLLO

En este capítulo se evidencian todas las fases del desarrollo haciendo uso de la metodología ágil Scrum, como se muestra en la Fig. 5.



Figura 5. Fases de desarrollo de Scrum

### 3.1 Desarrollo Etapa Pre-Juego

#### 3.1.1 Definición del equipo SCRUM

En la tabla 2 se muestra el equipo SCRUM establecido para el desarrollo de la aplicación web del cementerio indígena de Otavalo.

Tabla 2. Roles del Proyecto

Nombre	Rol	Cargo
Xavier Rea	Product Owner	Director del Trabajo de Grado Docente UTN
Verónica Saransig	Scrum Master Equipo de Desarrollo	Tesista
Organización UNORICO SAMASHUNCHIK	Stakeholder	Directiva de la Organización del cementerio

### 3.1.2 Elicitación de Requerimientos

Los requerimientos de la aplicación fueron determinados por medio de historias de usuario, basándose en los principios definidos por la metodología ágil SCRUM [54]. Para esto se llevó a cabo una reunión con la Directiva de la Organización UNORICO SAMASHUNCHIK del cementerio de Otavalo, en la que se analizaron y definieron todos los procesos esenciales para la gestión del cementerio.

*Tabla 3 Historia de Usuario 1*

<b>Historia de usuario</b>	
<b>ID: HUM-01</b>	Usuario: Usuarios
<b>Nombre:</b>	Iniciar Sesión
<b>Prioridad: Alta</b>	Dependencia: N/A
<b>Descripción:</b>	Como usuario de la aplicación quiero iniciar sesión con mi correo y contraseña para ingresar a la aplicación web y acceder a las funcionalidades de este
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe de mostrar una pantalla de inicio de sesión que contenga los campos correo y contraseña. <b>CA2:</b> Los campos correo y contraseña deben de ser obligatorios. En caso de que alguno se encuentre vacío debe de indicarse un mensaje de error. <b>CA3:</b> El campo de correo debe de verificar que sea un email válido y debe de indicar un mensaje mostrando de cómo debería de ser un correo correcto. <b>CA4:</b> Si el correo ingresado no existe debe de salir un mensaje “Usuario no existe”. <b>CA5:</b> Si la contraseña ingresada no es correcta debe de salir un mensaje “Credenciales Incorrectas”. <b>CA6:</b> Si al iniciar sesión es exitoso, el usuario es redirigido al panel del “Dashboard”. <b>CA7:</b> El campo contraseña debe de tener el icono del ojo que permite visualizar y ocultar la contraseña. <b>CA8:</b> Si un usuario intenta ingresar a alguna ruta sin autenticación será redirigido al login. <b>CA9:</b> La aplicación permite cerrar sesión en cualquier instante y se redirige a la página de login.

Tabla 4 Historia de Usuario 2

<b>Historia de usuario</b>	
<b>ID: HUM-02</b>	Usuario: Usuarios
<b>Nombre:</b>	Recuperar Contraseña
<b>Prioridad: Alta</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	<p>Como usuario de la aplicación quiero recuperar mi contraseña por medio de un enlace que se envíe a mi correo para lograr restablecer mi acceso a la aplicación web en caso de olvidarme o que se bloquee mi cuenta.</p> <p><b>CA1:</b> La pantalla de login debe de mostrar un enlace “Olvidaste tu contraseña?” que me redirija a la vista o página de recuperar contraseña.</p> <p><b>CA2:</b> La página de recuperar contraseña debe de tener un campo donde pueda ingresar el correo de manera obligatoria y diga de título “¿Olvidaste tu contraseña?”.</p> <p><b>CA3:</b> El campo del correo debe de validar que sea un correo valido. Si es un correo correcto el borde de este cambia de color verde en caso contrario le indica con un mensaje como es un correo válido.</p> <p><b>CA4:</b> Cuando se envía el formulario, la aplicación debe de enviar un enlace de restablecimiento al correo registrado</p> <p><b>CA5:</b> Si el correo no se encuentra registrado la aplicación debe de mostrar un mensaje de “No encontramos un usuario con ese correo”</p>
<b>Criterios de aceptación:</b>	<p><b>CA6:</b> El enlace que fue enviado al correo debe de redirigir a la vista para restablecer la contraseña.</p> <p><b>CA7:</b> La nueva contraseña debe de cumplir con ciertos requisitos de seguridad:</p> <ul style="list-style-type: none"> <li>• Mínimo 8 caracteres</li> <li>• Al menos una letra mayúscula</li> <li>• Al menos una letra minúscula</li> <li>• Al menos un número</li> <li>• Al menos un carácter especial (!@#\$.)</li> </ul> <p><b>CA8:</b> Los campos para la nueva contraseña deben de tener el icono para visualizar y ocultar la contraseña.</p> <p><b>CA9:</b> Se deben de confirmar que las contraseñas coincidan en caso de no hacerlo mostrar un mensaje “Las contraseñas no coinciden”.</p> <p><b>CA10:</b> Al restablecer la contraseña el contador de intentos fallidos vuelve a 0 y en caso de estar bloqueada la cuenta se desbloquea.</p> <p><b>CA11:</b> Cuando completa el restablecimiento de la contraseña es redirigido de vuelta a la página de login.</p>

Tabla 5 Historia de Usuario 3

<b>Historia de usuario</b>	
<b>ID: HUM-03</b>	Usuario: Usuarios autenticados
<b>Nombre:</b>	Gestionar Mi Perfil
<b>Prioridad: Media</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	Como usuario autenticado de la aplicación quiero ver y poder actualizar mi información personal para mantener mis datos personales actualizados correctamente.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe de tener un apartado “Mi perfil” ingresando se muestre la información del usuario como rol y nombre.</p> <p><b>CA2:</b> La página del perfil, debe de contar con un formulario que contiene los campos : Nombres completos, correo, locación, teléfono y un acerca de mí con datos anteriormente llenados.</p> <p><b>CA3:</b> Los campos nombres completos, correo son obligatorios mientras que los demás campos son opcionales.</p> <p><b>CA4:</b> El correo debe de ser único en la aplicación. En caso de que el usuario lo cambie por uno ya registrado le saldrá un mensaje de error.</p> <p><b>CA5:</b> Cuando se guarden los cambios realizados exitosamente, se debe de mostrar un mensaje “perfil actualizado correctamente”.</p> <p><b>CA6:</b> Cuando intente guardar los cambios realizados y ocurre un error debe de salir un mensaje “Error al actualizar perfil”.</p> <p><b>CA7:</b> Los mensajes de éxito y error deben solo de durar 3 segundos y desaparecer automáticamente una vez culminado el tiempo estimado.</p>

Tabla 6. Historia de Usuario 4

<b>Historia de usuario</b>	
<b>ID: HUM-04</b>	Usuario: Administrador / presidente
<b>Nombre:</b>	Gestión de usuarios
<b>Prioridad: Alta</b>	HUM-01 (Iniciar Sesión)

<b>Descripción:</b>	Como administrador quiero crear, editar y deshabilitar cuentas de los usuarios para controlar quien tendrá acceso y mantener la seguridad de la aplicación
	<p><b>CA1:</b> La aplicación debe de permitir crear un nuevo usuario desde una ventana (modal) con los campos: Nombres completos, correo, contraseña, teléfono, ubicación, rol y estado que se le otorga al usuario.</p> <p><b>CA2:</b> Los campos Nombres completos, Correo, Contraseña y Rol son obligatorios.</p> <p><b>CA3:</b> El correo electrónico debe de ser único. En caso de que ya se encuentre registrado debe de indicar “Este correo ya está registrado”.</p> <p><b>CA4:</b> El campo de correo debe de validar el formato de email valido. Indicando un mensaje de cómo es un correo correcto.</p> <p><b>CA5:</b> La contraseña debe de contar con ciertos requisitos como:</p> <ul style="list-style-type: none"> <li>• Mínimo 8 caracteres</li> <li>• Al menos una letra mayúscula</li> <li>• Al menos una letra minúscula</li> <li>• Al menos un número</li> <li>• Al menos un carácter especial (@\$!%*?&amp;)</li> </ul> <p><b>CA6:</b> El campo de contraseña debe de contar con el icono del ojo para mostrar u ocultar la contraseña.</p> <p><b>CA7:</b> El campo de teléfono debe de solo aceptar números con un máximo de 10 dígitos.</p>
<b>Criterios de aceptación:</b>	<p><b>CA8:</b> El código del Usuario se debe de generar automáticamente cuando se crea uno nuevo.</p> <p><b>CA9:</b> Cuando se crea un nuevo usuario exitosamente se debe de indicar un mensaje “El usuario xxxx fue creado exitosamente”.</p> <p><b>CA10:</b> La aplicación debe permitir editar los datos de un usuario registrado.</p> <p><b>CA11:</b> Cuando se edite el correo debe de seguir siendo único en cuanto a los demás usuarios.</p> <p><b>CA12:</b> Una vez actualizados los datos se muestra un mensaje “El usuario xxxx fue actualizado correctamente”.</p> <p><b>CA13:</b> Es necesario permitir el activar o desactivar una la cuenta de un usuario mediante un toggle. Para no eliminarlo y no perder su historial de actividades.</p> <p><b>CA14:</b> La página principal debe de indicar un listado de los usuarios existentes en una tabla con los siguientes datos: Nombres, correo, rol y estado con su respectiva paginación.</p> <p><b>CA15:</b> El buscador tiene que permitir filtrar usuarios por nombres, correo o su código único.</p> <p><b>CA16:</b> La aplicación debe generar reportes de todos los usuarios o seleccionar algunos en formato PDF e Excel</p>

---

con todos los campos necesarios: Nombres, correo, teléfono, ubicación, rol y su estado.

---

**CA17:** Si no se selecciona ningún usuario cuando se intenta generar el reporte mostrar un mensaje “Debe de seleccionar al menos un usuario”.

---

*Tabla 7. Historia de Usuario 5*

---

<b>Historia de usuario</b>	
<b>ID: HUM-05</b>	Usuario: Administrador / presidente
<b>Nombre:</b>	Gestión de Roles y Permisos
<b>Prioridad: Alta</b>	Dependencia: HUM-04 (Gestión de Usuarios)
<b>Descripción:</b>	Como administrador quiero crear roles personalizados y asignarles permisos específicos (Ver, Crear, Editar, Eliminar), para controlar estrictamente qué funciones puede ejecutar cada tipo de usuario.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> El sistema debe permitir crear un nuevo Rol (ej. "Secretario") desde una ventana (modal) que tenga el campo nombre del rol y debe de ser obligatorio.</p> <p><b>CA2:</b> El nombre del rol debe de ser único caso contrario mostrar un mensaje de error.</p> <p><b>CA3:</b> El código único se genera automáticamente cuando se genera un nuevo registro.</p> <p><b>CA4:</b> Al crear el rol exitosamente indicar un mensaje “El rol xxxx ha sido creado correctamente”.</p> <p><b>CA5:</b> La aplicación tiene que permitir editar el nombre de un rol existente desde un modal.</p> <p><b>CA6:</b> No debe permitirse renombrar el rol “Administrador”. Si trata de hacerlo mostrar “No se puede renombrar el rol Administrador”.</p> <p><b>CA7:</b> Al actualizar el nombre de un rol exitosamente indicar “Rol xxxx actualizado correctamente”.</p> <p><b>CA8:</b> La aplicación debe de permitir eliminar un rol mediante una confirmación previa.</p> <p><b>CA9:</b> No se puede eliminar el rol “Administrador”. Si hace el intento mostrar “No se puede eliminar, existe usuarios utilizando este rol”.</p> <p><b>CA10:</b> Al crear el rol exitosamente indicar un mensaje “El rol xxxx ha sido creado correctamente”.</p> <p><b>CA11:</b> Cuando se eliminar un rol exitosamente presentar un mensaje “El rol xxxx ha sido eliminado correctamente”.</p> <p><b>CA12:</b> La aplicación debe de indicar un apartado de permisos para cada rol existente.</p>

---

	<b>CA13:</b> Cada uno de los permisos debe de mostrarse con un checkbox el cual se puede marcar o desmarcar individualmente.
	<b>CA14:</b> Cada grupo de permisos para cada rol debe de tener los botones “Todos” o “Ninguno” para seleccionar o deseleccionar todos los permisos en masa.
	<b>CA15:</b> Una vez guardado los cambios se debe de mostrar un mensaje “Los permisos han sido actualizados correctamente”.
	<b>CA16:</b> El listado en la tabla de roles debe de indicar: código, nombre y los permisos asignados que tiene cada rol con su respectiva paginación.

*Tabla 8. Historia de Usuario 6*

<b>Historia de usuario</b>	
<b>ID: HUM-06</b>	Usuario: Administrador / presidente
<b>Nombre:</b>	Gestión de Ubicaciones Geográficas
<b>Prioridad: Alta</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	Como administrador quiero gestionar los Cantones, Parroquias y Comunidades u Organizaciones, para poder relacionar a los socios con el lugar de origen.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación tiene que permitir crear, editar eliminar y buscar cantones. El campo nombre debe ser obligatorio y se convierte automáticamente en mayúsculas.
	<b>CA2:</b> El nombre el cantón debe ser único. Si se reitera mostrar “Este nombre del cantón ya está registrado”.
	<b>CA3:</b> El código del cantón se genera automáticamente cuando se registra o crea uno nuevo.
	<b>CA4:</b> No es posible eliminar un cantón que tenga parroquias vinculadas. En caso de hacerlo indicar “No se puede eliminar tiene parroquias asociadas”.
	<b>CA5:</b> El buscar debe de filtrar por nombre o código del Cantón.
	<b>CA6:</b> La aplicación debe permitir crear, editar, eliminar y buscar parroquias. El campo nombre tiene que ser obligatorio y convertido en mayúsculas.
	<b>CA7:</b> El nombre de la parroquia debe ser único dentro del mismo cantón. Si se repite presentar “Ya existe una parroquia con ese nombre en el cantón seleccionado”.
	<b>CA8:</b> El código de la parroquia se genera automáticamente.

<b>CA9:</b>	La tabla donde está el listado de las parroquias tiene que filtrar por cantón por medio un select desplegable.
<b>CA10:</b>	El buscar debe de filtrar tanto por nombre como por el código único de la parroquia.
<b>CA11:</b>	Se tiene que generar reportes en formato PDF e Excel de todas las parroquias o solo de las seleccionadas.
<b>CA12:</b>	La aplicación debe permitir crear, editar, eliminar y buscar comunidades u organizaciones. El campo nombre tiene que ser obligatorio.
<b>CA13:</b>	El nombre de la comunidad tiene que ser único dentro de la misma parroquia. En caso de repetirse mostrar “La comunidad xxxx ya existe en la parroquia xxxx”.
<b>CA14:</b>	El código único de la comunidad se genera automáticamente.
<b>CA15:</b>	La tabla donde se encuentra el listado de registro debe permitir el filtrado por parroquias a través de un select desplegable.
<b>CA16:</b>	Se debe de generar reportes en formato PDF y Excel de todas las comunidades o de ciertas seleccionadas.
<b>CA17:</b>	Cuando se seleccione la ubicación en los otros formularios como socio u otros las parroquias tienen que cargarse dinámicamente. Lo mismo sucede cuando selecciono una parroquia las comunidades se cargan automáticamente.

*Tabla 9 Historia de Usuario 7*

<b>Historia de usuario</b>	
<b>ID: HUM-07</b>	Usuario: Administrador / presidente
<b>Nombre:</b>	Gestión de Servicios
<b>Prioridad: Media</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	Como administrador quiero gestionar el apartado de servicios que brinda el cementerio para que se encuentre disponibles al instante de realizar las facturas.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe permitir crear un servicio con los campos: nombre (obligatorio) , descripción (opcional), precio (numérico).</p> <p><b>CA2:</b> El código único de servicio se genera automáticamente cuando se crea un nuevo servicio,</p>

	<b>CA3:</b> La aplicación debe permitir editar un servicio modificando todos los campos que se encuentran.
	<b>CA4:</b> La aplicación debe de permitir eliminar un servicio. Cuando se elimina mostrar un mensaje “Servicio Eliminado”.
	<b>CA5:</b> El buscar tiene que filtrar por nombre o código en tiempo real.
	<b>CA6:</b> Se debe de seleccionar todos los servicios o algunos para generar un reporte en formato PDF y Excel.
	<b>CA7:</b> En caso de no seleccionar ningún servicio y tratar de generar el reporte se indicará un mensaje “Debe de seleccionar al menos un servicio”.

*Tabla 10 Historia de Usuario 8*

<b>Historia de usuario</b>	
<b>ID: HUM-08</b>	Usuario: Administrador / presidente
<b>Nombre:</b>	Gestión de Beneficios
<b>Prioridad: Media</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	Como administrador quiero gestionar el apartado de beneficios que pueden obtener los socios para que se encuentre disponibles al instante de realizar las facturas.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe permitir crear un beneficio con los campos: nombre (obligatorio), descripción (opcional), tipo (obligatorio) y valor ( obligatorio, numérico).
	<b>CA2:</b> El código único de beneficios se genera automáticamente cuando se crea un nuevo beneficio.
	<b>CA3:</b> La aplicación debe permitir editar un beneficio modificando todos los campos que se encuentran.
	<b>CA4:</b> La aplicación debe de permitir eliminar un beneficio. Cuando se elimina mostrar un mensaje “Beneficio Eliminado”.
	<b>CA5:</b> El buscar tiene que filtrar por nombre o código en tiempo real.
	<b>CA6:</b> Se debe de seleccionar todos los beneficios o algunos para generar un reporte en formato PDF y Excel.
	<b>CA7:</b> En caso de no seleccionar ningún beneficio y tratar de generar el reporte se indicará un mensaje “Debe de seleccionar al menos un beneficio”.

Tabla 11. Historia de Usuario 9

<b>Historia de usuario</b>	
<b>ID: HUM-09</b>	Usuario: Administrador / presidente / secretario
<b>Nombre:</b>	Gestión de Socios
<b>Prioridad: Alta</b>	Dependencia: HUM-06 (Ubicaciones Geográficas)
<b>Descripción:</b>	<p>Como administrador quiero registrar, editar, visualizar y eliminar socios del cementerio con todos sus datos personales para preservar un padrón actualizado.</p> <p><b>CA1:</b> La aplicación tiene que permitir crear un socio con los siguientes campos: Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono, Dirección, Email, Comunidad, Estado Civil, Género, Fecha de Inscripción, Tipo de Beneficio (Sin Subsidio / Con Subsidio / Exonerado), Condición (Ninguna / Discapacidad / Enfermedad Terminal) y Estatus (Vivo / Fallecido).</p> <p><b>CA2:</b> Se debe de tomar en cuenta que estos campos son obligatorios: Cédula, Nombres, Apellidos, Fecha de Nacimiento, Comunidad, Estado Civil, Fecha de Inscripción, Tipo de Beneficio, Condición y Estatus.</p> <p><b>CA3:</b> La cédula debe ser única. En caso de repetirse mostrar el mensaje “Este número de cédula ya existe. El socio ya esa registrado”.</p> <p><b>CA4:</b> Si el tipo de beneficio es “Exonerado” debe de tomarse en cuenta que la edad del socio sea mayor o igual a 75 años. Si no cumple con esto mostrar el mensaje “No se puede crear como exonerado”.</p> <p><b>CA5:</b> Si el tipo de beneficio es “Exonerado” y cumple con el requisito de mayor o igual a 75 años se debe de poner obligatoriamente la fecha de exoneración.</p> <p><b>CA6:</b> El código del socio se genera automáticamente cuando se registra.</p> <p><b>CA7:</b> Cuando se crea exitosamente indicar el mensaje “Socio creado correctamente xxxx”.</p> <p><b>CA8:</b> La aplicación debe permitir editar todos los campos del socio</p> <p><b>CA9:</b> La validación del beneficio “Exonerar” por edad mínima 75 años también se puede editar.</p> <p><b>CA10:</b> Cuando se actualiza exitosamente los datos de un socio debe de mostrar un mensaje “Socio actualizado correctamente”.</p> <p><b>CA11:</b> Cuando se ve el detalle de un socio se debe de indicar todos sus datos personales y los nichos que este tiene con sus respectivas ubicaciones (bloque / nicho).</p>
<b>Criterios de aceptación:</b>	

---

**CA12:** La aplicación debe de solicitar una confirmación antes de eliminar un socio.

---

**CA13:** Al eliminar exitosamente se muestra un mensaje “Socio eliminado correctamente”.

---

*Tabla 12. Historia de Usuario 10*

---

<b>Historia de usuario</b>	
<b>ID: HUM-10</b>	Usuario: Administrador / secretario / presidente
<b>Nombre:</b>	Buscar y Filtrar Socios
<b>Prioridad: Media</b>	Dependencia: HUM-09 (Gestionar Socios)
<b>Descripción:</b>	Como administrador quiero buscar y filtrar socios por diferentes criterios en el listado general, para localizarlos rápidamente la información que necesito sin revisar página por página.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe tener una barra de búsqueda que filtre en tiempo real por: Cédula, Apellidos, Nombres o Código del socio.
	<b>CA2:</b> Los resultados de la búsqueda deben de actualizarse dinámicamente, cuando el usuario se encuentre escribiendo con un icono de carga.
	<b>CA3:</b> La aplicación tiene que permitir filtrar a los socios por comunidades por medio de un select desplegable.
	<b>CA4:</b> La aplicación debe permitir filtrar por estatus seleccionado (vivo o fallecido).
	<b>CA5:</b> La aplicación tiene que permitir filtrar por tipo de beneficio escogiendo varias opciones (Subsidio, Sin subsidio, Exonerado).
	<b>CA6:</b> La aplicación debe filtrar por mes y año de inscripción o registro.
	<b>CA7:</b> La tabla donde se encuentra el listado debe de tener los campos: código, cédula, nombres y apellidos, comunidad, tipo de beneficio, estatus, y el conteo de nichos (total, propios y compartidos).
	<b>CA8:</b> Los registros deben de estar con una paginación de 10 por página, conservando los filtros cuando se cambie de página.

---

Tabla 13. Historia de Usuario 11

<b>Historia de usuario</b>	
<b>ID: HUM-11</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Control de Exonerados
<b>Prioridad: Media</b>	Dependencia: HUM-09 (Gestionar Socios)
<b>Descripción:</b>	Como administrador quiero que la aplicación identifique automáticamente a ciertos socios que tienen la edad de 75 años o más que aún no se encuentran exonerados para administrar oportunamente su cambio de beneficio.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación nos indica una alerta en el apartado de gestión de socios cuando exista personas para exoneración (socios con 75 años o más que aún no tengan el beneficio de exonerado).
	<b>CA2:</b> Esta alerta debe de mostrar la cantidad de socios que han sido identificados para exonerar.
	<b>CA3:</b> Al crear o editar un socio con el beneficio “Exonerado” se debe de validar que cumpla con el requisito “mayor o igual a 75 años” caso contrario muestra un mensaje “No se puede crear como exonerado”.
	<b>CA4:</b> Cuando se el beneficio se Exonerado y cumpla con el requisito la fecha de exoneración es obligatorio.
	<b>CA5:</b> La aplicación debe de calcular la edad del socio automáticamente con la fecha de nacimiento.
	<b>CA6:</b> Se tiene que poder filtrar los socios por el estado de beneficio “Exonerado” para visualizar cuales son los socios exonerados,

Tabla 14. Historia de Usuario 12

<b>Historia de usuario</b>	
<b>ID: HUM-12</b>	Usuario: Administrador / secretario / presidente
<b>Nombre:</b>	Reporte de Socios
<b>Prioridad: Baja</b>	Dependencia: HUM-10 (Buscar Socios)
<b>Descripción:</b>	Como administrador quiero generar reportes en formato PDF o Excel de los socios, para tener el listado impreso del padrón.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación tiene que permitir seleccionar todos lo algunos socios mediante checkboxes en la tabla.

	<b>CA2:</b> Se debe de poder generar un reporte en formato PDF o Excel con sus respectivos campos.
	<b>CA3:</b> El reporte en formato PDF debe de ser en formato A4 horizontal para visualizar correctamente todos los datos.
	<b>CA4:</b> Los filtros que se tiene en la tabla como estatus y beneficio al momento de generar el reporte deben de aplicarse.
	<b>CA5:</b> En caso de filtrar por mes / año en el reporte se debe de poner un subtítulo indicando el tipo de filtro que se aplicó.
	<b>CA6:</b> Sino se selecciona ningún socio se muestra un mensaje “Debe seleccionar al menos un socio para generar el reporte”.

*Tabla 15. Historia de Usuario 13*

<b>Historia de usuario</b>	
<b>ID: HUM-13</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Gestión de Bloques
<b>Prioridad: Alta</b>	Dependencia: HUM-01 (Iniciar Sesión)
<b>Descripción:</b>	Como administrador quiero gestionar los bloques (sectores) del cementerio para organizar la distribución de los nichos relacionándolos por medio de polígonos de GIS.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe permitir crear un Bloque con los campos: Nombre (obligatorio), Descripción (opcional), Área m<sup>2</sup> (opcional) y polígono del GIS (opcional).</p> <p><b>CA2:</b> Cuando selecciono un polígono GIS el código del bloque se coordina automáticamente con el del GIS y solo se indican los que se encuentren disponibles (Sin asignación).</p> <p><b>CA3:</b> Si no se ingresa el área manualmente pero se ha seleccionado un polígono GIS este lo calcula automáticamente y desde la geometría del mapa.</p> <p><b>CA4:</b> No se puede asignar un polígono GIS que ya está registrado con otro bloque.</p> <p><b>CA5:</b> La aplicación tiene que permitir editar, eliminar bloques.</p> <p><b>CA6:</b> El buscar debe de filtrar por código, nombre del bloque y con su respectiva paginación.</p> <p><b>CA7:</b> Se puede generar reportes en formato PDF y Excel con todos los campos.</p>

Tabla 16. Historia de Usuario 14

<b>Historia de usuario</b>	
<b>ID: HUM-14</b>	Usuario: Administrador / secretario / presidente
<b>Nombre:</b>	Gestión de Nichos
<b>Prioridad: Alta</b>	Dependencia: HUM-13 (Gestión de Bloques), HUM-09 (Gestionar Socios)
<b>Descripción:</b>	Como administrador quiero gestionar los nichos del cementerio para supervisar la ubicación, capacidad, estado, que tipo es, y su disponibilidad.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe permitir crear un nicho con los campos: Bloque (obligatorio), Socio responsable (obligatorio), Tipo (PROPIO/COMPARTIDO, obligatorio), Clase (BÓVEDA/TIERRA, obligatorio), Capacidad (obligatorio, mínimo 1), Estado físico (Bueno, malo, mantenimiento)
	<b>CA2:</b> Cuando se crea un nuevo nicho la ocupación comienza en 0 y la disponibilidad en verdadero.
	<b>CA3:</b> Si se desea se puede vincular con un polígono GIS haciendo que el código se sincronice automáticamente con el código del mapa de igual forma solo se muestran los que no han sido asignados.
	<b>CA4:</b> La aplicación tiene que permitir editar, eliminar nichos.
	<b>CA5:</b> El buscar debe filtrar por nombres, código, cedula y nombres de un socio. También tiene que filtrar por bloque y estado físico.
	<b>CA6:</b> Tiene que generar un código QR para cada uno de los nichos en modo texto (datos del nicho) este se puede previsualizar y descargar en PNG.
	<b>CA7:</b> Debe de generar reportes PDF y Excel con todos sus campos.

Tabla 17. Historia de Usuario 15

<b>Historia de usuario</b>	
<b>ID: HUM-15</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Ver Mapa Cementerio
<b>Prioridad: Media</b>	Dependencia: HUM-13 (Gestión de Bloques), HUM-14 (Gestión de Nichos)

<b>Descripción:</b>	Como administrador quiero ver un mapa interactivo del cementerio para lograr identificar la ubicación de los bloques y sus nichos.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe de mostrar un mapa interactivo que se visualice los bloques, nichos del cementerio.</p> <p><b>CA2:</b> Al seleccionar o dar clic en un nicho este debe de abrir una ventana mostrando información como: Estado, Bloque, sector, información dentro del nicho.</p> <p><b>CA3:</b> El mapa debe de ser interactivo permitiendo hacer zoom y moverse en el área del terreno.</p> <p><b>CA4:</b> La vista del mapa debe de estar integrada en el apartado asignaciones en la vista detalle donde ya se encuentra la asignación del nicho, socio y fallecido para poder visualizar y que nos lleve directo al nicho del fallecido.</p>

*Tabla 18. Historia de Usuario 16*

<b>Historia de usuario</b>	
<b>ID: HUM-16</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Gestión de Fallecidos
<b>Prioridad: Alta</b>	Dependencia: HUM-06 (Ubicaciones Geográficas)
<b>Descripción:</b>	Como administrador quiero registrar, editar y eliminar la información de las personas fallecidas para tener un registro actualizado y asignarlo a un nicho.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe permitir registrar a un fallecido con los siguientes campos: Cédula (obligatoria, única), Nombres (obligatorio), Apellidos (obligatorio), Fecha de Nacimiento (obligatorio), Fecha de Fallecimiento (obligatorio), Comunidad (obligatorio), Género (opcional), Estado Civil (opcional) y Observaciones (obligatorio).</p> <p><b>CA2:</b> Cuando se ingresa una cedula que coincide con un socio, se autocompletan los datos automáticamente.</p> <p><b>CA3:</b> Si registra un fallecido y la cedula coincide con la de un socio el estatus de este en la gestión socios cambia automáticamente a “Fallecido”.</p>

---

**CA4:** El código del fallecido se genera automáticamente

**CA5:** La aplicación tiene que permitir editar y eliminar registros de fallecidos.

**CA6:** Cuando se registra exitosamente a un fallecido debe de indicar un mensaje “Fallecido registrado correctamente”.

---

*Tabla 19. Historia de Usuario 17*

---

<b>Historia de usuario</b>	
<b>ID: HUM-17</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Buscar Fallecidos
<b>Prioridad: Alta</b>	Dependencia: HUM-16 (Gestión de Fallecidos)
<b>Descripción:</b>	Como administrador quiero buscar y filtrar a los fallecidos que se encuentran registrados para ubicar su información rápidamente.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe contar con una barra de búsqueda que filtre por cédula, nombres, apellidos o el código de la persona fallecida.
	<b>CA2:</b> La aplicación tiene que permitir filtrar por comunidad por medio de un select desplegable.
	<b>CA3:</b> La aplicación debe permitir filtrar por año y mes de los fallecidos.
	<b>CA4:</b> Los resultados deben de estar con una paginación de 10 registros por página conservando los filtros cuando se cambie de página.
	<b>CA5:</b> El listado en la tabla debe de indicar: código, cédula, nombres, apellidos, comunidad, fecha de nacimiento, fecha de fallecimiento.

---

*Tabla 20. Historia de Usuario 18*

---

<b>Historia de usuario</b>	
<b>ID: HUM-18</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Inhumación (Asignar)
<b>Prioridad: Alta</b>	Dependencia: HUM-14 (Gestión de Nichos), HUM-16 (Gestión de Fallecidos), HUM-09 (Gestionar Socios)

---

<b>Descripción:</b>	Como administrador quiero asignar personas fallecidas a un nicho del cementerio igualmente asociarlo a un socio responsable para supervisar el control de ocupación de nichos.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación debe permitir crear una asignación seleccionando: Nicho (obligatorio) solo se desplegarán nichos disponibles (sin asignación), Socio responsable (obligatorio), Fallecido (obligatorio) tomando en cuenta solo a los socios sin asignación, rol del socio y fecha de inhumación de manera obligatoria.
	<b>CA2:</b> La aplicación tiene que verificar la capacidad del nicho. En caso de estar al límite mostrar un mensaje “EL nicho se encuentra al límite”.
	<b>CA3:</b> El código de asignación se genera de manera automática con el formato ASG-XX
	<b>CA4:</b> Cuando se asigna, la ocupación del nicho se incrementa automáticamente y la disponibilidad se actualiza.
	<b>CA5:</b> La aplicación debe permitir cambiar al socio responsable, fallecido y la fecha de inhumación y su rol.
	<b>CA6:</b> La aplicación tiene que permitir eliminar una asignación equivocada. Entonces la ocupación del nicho se decrementa automáticamente.
	<b>CA7:</b> El listado en la tabla asignaciones debe de mostrar fallecidos, socios, ocupación (1/3), estado del nicho.
	<b>CA8:</b> El buscar tiene que filtrar por código de nicho, nombres, cédula del socio o del fallecido. También se puede filtrar mediante estado del nicho y mes/año de inhumación.

*Tabla 21. Historia de Usuario 19*

<b>Historia de usuario</b>	
<b>ID: HUM-19</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Exhumación
<b>Prioridad: Alta</b>	Dependencia: HUM-18 (Inhumación / Asignaciones)
<b>Descripción:</b>	Como administrador quiero registrar la exhumación (retirar restos) de un fallecido de un nicho para conservar un historial completo del difunto.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación tiene que indicar un formulario para exhumar a un fallecido del nicho seleccionado. No puede exhumar si ya está exhumado solo a los que no lo están.

	<p><b>CA2:</b> Si el nicho no tiene ocupantes (fallecidos exhumados) debe indicar un mensaje “Este nicho no tiene ocupantes activos (fallecidos sin exhumar)”.</p> <p><b>CA3:</b> Cuando se exhuma se debe de registrar la fecha en la cual se realizó la exhumación (obligatorio) y colocar una observación.</p> <p><b>CA4:</b> Una vez registrada la exhumación la ocupación de nichos decreta automáticamente y la disponibilidad se recalcula.</p> <p><b>CA5:</b> La aplicación debe permitir generar reportes en PDF y Excel de todas las asignaciones.</p> <p><b>CA6:</b> La aplicación debe permitir generar reporte en formato PDF de todo los exhumados.</p>
--	---

*Tabla 22. Historia de Usuario 20*

<b>Historia de usuario</b>	
<b>ID: HUM-20</b>	Usuario: Administrador / secretario
<b>Nombre:</b>	Reporte de Asignaciones
<b>Prioridad: Media</b>	Dependencia: HUM-18 (Gestión de Asignaciones)
<b>Descripción:</b>	Como administrador quiero generar reportes en formato PDF y Excel de todas las asignaciones que se encuentran registrados para tener archivos de manera digital e impresa.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación generar un reporte en formato PDF y Excel de todas las asignaciones registradas.</p> <p><b>CA2:</b> Tiene que generar los reportes en PDF o Excel con los campos: ID, código, nombre del fallecido, fecha de inhumación, nombre del responsable, nicho, estado, ocupación.</p> <p><b>CA3:</b> El reporte PDF debe de estar en formato A4 horizontal.</p> <p><b>CA4:</b> Cuando se filtre por mes/año el reporte debe de tener un subtítulo indicando que filtros se aplicaron.</p> <p><b>CA5:</b> La aplicación debe permitir generar un reporte en formato PDF solo de exhumaciones.</p>

*Tabla 23. Historia de Usuario 21*

<b>Historia de usuario</b>	
<b>ID: HUM-21</b>	Usuario: Administrador / secretario

<b>Nombre:</b>	Ver Deudas
<b>Prioridad: Alta</b>	Dependencia: HUM-09 (Gestionar Socios), HUM-24 (Registrar Pagos)
<b>Descripción:</b>	Como administrador quiero ver el estado de deuda de los socios (años pendientes de pago) para saber quiénes son los que deben y proceder al cobro.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe calcular los años de pago pendientes automáticamente desde la inscripción del socio hasta la fecha actual.</p> <p><b>CA2:</b> En la vista gestión de pagos, se debe de indicar un resumen del socio indicando el estado de la cuenta: en color rojo si tiene años pendientes y en verde que se encuentra al día.</p> <p><b>CA3:</b> En la vista gestión de pagos debe de verse el historial de pagos que ha realizado el socio y el registrar nuevo pago con los años pendientes.</p> <p><b>CA4:</b> La sección de años pendientes solo muestra los años pendientes cuando existen en caso de no tener muestra un mensaje de “No tiene deudas”.</p> <p><b>CA5:</b> Al dar clic en el botón show se abrirá una ventana para ver los detalles del historial de pago del socio detallando años pagados, pendientes y su información.</p> <p><b>CA6:</b> La venta de historial de pagos debe de tener un botón “Ir a cobrar” en cada socio para dirigirse al cobro.</p> <p><b>CA7:</b> Para acceder al historial del socio debe de estar protegido por permisos.</p> <p><b>CA8:</b> Para crear un pago y ver deben de estar con permisos para acceder o ejecutar estas acciones.</p> <p><b>CA9:</b> La lista debe estar ordenada cronológicamente por fecha de fallecimiento (del más reciente al más antiguo) o alfabéticamente.</p>

*Tabla 24. Historia de Usuario 22*

<b>Historia de usuario</b>	
<b>ID: HUM-22</b>	Usuario: Administrador / tesorero
<b>Nombre:</b>	Facturación
<b>Prioridad: Media</b>	Dependencia: HUM-09 (Socios), HUM-07 (Servicios), HUM-08 (Beneficios)
<b>Descripción:</b>	Como administrador o tesorero encargado del cementerio quiero crear, editar, emitir y anular facturas cuando se trata de beneficios y servicios brindados para conservar un registro de los ingresos y crear comprobantes en formato PDF para los socios o personas externas.

---

**Criterios de aceptación:**

**CA1:** La aplicación debe permitir crear una factura escogiendo un socio registrado por medio del buscar en tiempo real ya se cedula o nombres y apellidos de igual manera si o es un socio se puede ingresar los datos de un cliente externo de forma manual.

**CA2:** Cuando seleccionamos un socio registrado los campos se llenan automáticamente con los datos de él.

**CA3:** El campo nombre y apellidos son obligatorio para el cliente, los demás campos pueden ser opcionales.

**CA4:** Cuando se realice la factura obligatoriamente debe estar la fecha de emisión y por defecto se establece la fecha actual.

**CA5:** La aplicación tiene que permitir agregar múltiples líneas de detalles. Cada una de las líneas requiere seleccionar un Tipo que puede varias entre Beneficio o Servicio, después debe de seleccionar un ítem del catálogo correspondiente, cantidad y precio unitario.

**CA6:** Cuando se selecciona un ítem del catálogo, el precio de este debe llenarse automáticamente con el valor predeterminado ya sea el beneficio o servicio.

**CA7:** El subtotal de cada línea y el total deben se calcularse en tiempo real.

**CA8:** La factura por lo menos debe tener un ítem en detalle si no hay mostrar un mensaje “Debe agregar al menos un ítem”.

**CA9:** El código único debe generarse automáticamente en formato FAC-00001 de manera consecutiva.

**CA10:** Cuando se guarda la factura exitosamente se crea en el estado de “borrador” y muestra un mensaje “Factura FAC-xxxx guardada como borrador” con esto puede editar o emitirla.

**CA11:** Las facturas que se encuentran estado de borrador se puede editar. Si la factura ya está emitida o anulada ya no permite editar.

**CA12:** Cuando se edita se debe de poder modificar los datos del cliente, su fecha y los ítems del detalle (agregar, retirar, cambiar cantidades).

**CA13:** Cuando se actualizan los campos y ítems se cambian completamente por los nuevos datos. El total y subtotal deben de recalcularse.

**CA14:** Al actualizar exitosamente se debe indicar un mensaje “Factura actualizada correctamente”.

**CA15:** Cuando se presione el botón para “Generar PDF”, la aplicación lanza una alerta de confirmación (SweetAlert) indicando que la factura pasara a estado “Emitida” y no se puede deshacer.

**CA16:** Cuando se confirma la emisión de la factura, se registra la fecha de emisión y el usuario que lo hizo y se descarga automáticamente en archivo PDF.

---

	<b>CA17:</b> Una factura ya emitida no se puede editar. Solo permite imprimir el PDF, ver los detalles o anular.
	<b>CA18:</b> Se puede anular una factura si está en el estado de borrador o emitida pero no se puede anular si ya fue anulada.
	<b>CA18:</b> Cuando se anula una factura, la aplicación solicita obligatoriamente un motivo por medio de un cuadro de texto. En caso de no dar el motivo sale un mensaje “Debe especificar el motivo de la anulación”.
	<b>CA19:</b> Cuando se anula se registra la fecha de anulación, el usuario que lo hizo y el motivo. Se muestra el mensaje “La FAC-XXXX ha sido anulada”.
	<b>CA20:</b> Una vez anulada la factura no se puede editar ni emitir. Solo permite ver el motivo de anulación por el botón “Motivo” que se muestra mediante un cuadro de información.
	<b>CA21:</b> Las facturas no se puede eliminar por políticas de seguridad y auditoría.
	<b>CA22:</b> En la tabla de la gestión de Facturas se debe de ver el listado el cual cuenta con las columnas: número de fila, código, fecha, cliente/socio (nombre + cédula), total y estado de color (amarillo = EN PROCESO, celeste = EMITIDA, rojo = ANULADA).
	<b>CA23:</b> La aplicación debe de tener una barra de buscar en tiempo real mediante código de factura, nombres del socio o cliente y cédula.
	<b>CA24:</b> El acceso a este apartado debe de estar controlado mediante roles y permisos.

*Tabla 25. Historia de Usuario 23*

<b>Historia de usuario</b>	
<b>ID: HUM-23</b>	Usuario: Administrador / tesorero
<b>Nombre:</b>	Registrar Pago
<b>Prioridad: Media</b>	Dependencia: HUM-09 (Gestionar Socios), HUM-23 (Ver Deudas)
<b>Descripción:</b>	Como administrador o tesorero quiero registrar los pagos anuales de los socios para tener su historial de pagos actualizado y correctamente gestionado.
<b>Criterios de aceptación:</b>	<b>CA1:</b> La aplicación cuando se va a registrar un pago debe de abrir una ventana para buscar al socio. <b>CA2:</b> El buscar debe de filtrar a los socios en tiempo real por medio de nombres, apellidos y cédula.

---

**CA3:** Una vez seleccionado al socio debe de dar clic en él o n un botón de cobrar que redirija a la ventana gestión de pagos.

**CA4:** En caso de no existir un socio con ciertos datos de búsqueda debe de salir “No se encontró ningún socio con esos datos”

**CA5:** Cuando estamos en la vista gestión de pagos del socio debe de haber dos secciones una para los años pendientes y el otro historial de los pagos que haya realizado

**CA6:** La cabecera de la vista gestión pagos debe de mostrar los datos del socio y como está su estado.

**CA7:** En la parte donde se encuentran los años pendientes deben de estar con checkboxes individuales. Donde el usuario debe seleccionar los años que debe de cobrar.

**CA8:** En caso de que el socio no tenga años pendientes en esa sección sale un cuadro con el mensaje “No tiene deudas pendientes”.

**CA9:** El campo fecha pago es obligatorio y por defecto se establece la fecha actual.

**CA10:** El campo observación es opcional lo que permite agregar una nota.

**CA11:** Debe se seleccionar un año pendiente para guardar el pago caso contrario indica un mensaje “Debe de seleccionar al menos un año para guardar el pago”.

**CA12:** Cuando se va a guardar un pago la aplicación verifica si existe un recibo del socio en la misma fecha (en el mismo día). En caso de no existir se crea un nuevo registro y se relaciona los pagos relacionados y si existe se agregan los nuevos años al registro existente.

**CA13:** No se permite registrar un año que ya fue cancelado.

**CA14:** El total del historial de pagos se recalcula automáticamente sumando los nuevos pagos realizados (suma de todos los registros).

**CA15:** Cuando se guarda correctamente, debe mostrar un mensaje “Pago registrado correctamente”.

**CA16:** La aplicación debe indicar un listado con paginación de 10 por página.

**CA17:** El listado de la tabla debe contar con las siguientes columnas: # Recibo, Socio (nombre + cédula), Años Cancelados (de color azul con los años), Fecha, Total y Acciones.

**CA18:** El buscar debe de filtrar por cédula, nombre y apellido del socio

**CA19:** La aplicación permite corregir los años que fueron asignados a un registro.

---

	<b>CA20:</b> Solo se indican los años que son menores al año actual (no permite años futuros).
	<b>CA21:</b> Se puede modificar el campo de observación del registro.
	<b>CA22:</b> Cuando se guarda el registro editado, se corrige los pagos realizados y se crean los nuevos pagos seleccionados, se recalcula el total del registro y muestra un mensaje “Registro Actualizado”.
	<b>CA23:</b> La aplicación no debe de eliminar los registros que ya han sido creados.
	<b>CA24:</b> El acceso para este apartado debe de estar controlado por roles y permisos para ejecutar todas las operaciones que se realiza aquí.

*Tabla 26. Historia de Usuario 24*

<b>Historia de usuario</b>	
<b>ID: HUM-24</b>	Usuario: Administrador / auditor
<b>Nombre:</b>	Auditoría (Logs)
<b>Prioridad: Media</b>	Dependencia: Todos los módulos funcionales
<b>Descripción:</b>	Como administrador o auditor de la aplicación quiero tener un registro detallado de todos los cambios realizados en la aplicación para monitorear la seguridad , transparencia de todos los datos y detectar cualquier error o acción.
<b>Criterios de aceptación:</b>	<p><b>CA1:</b> La aplicación debe registrar automáticamente las acciones realizadas por el usuario cada vez que crea, edita o elimina información en cualquier parte de la aplicación.</p> <p><b>CA2:</b> Cada acción realizada, se debe de obtener obligatoriamente: el usuario que la realizo, la acción efectuada, a que modulo afecto, el dato que afecto y la fecha que fue hecho.</p> <p><b>CA3:</b> En caso de ediciones, la aplicación debe indicar que información se tenía antes y como quedo después del cambio.</p> <p><b>CA4:</b> Los registros deben de ser claros y sencillos de leer para el administrador o auditor.</p> <p><b>CA5:</b> La aplicación tiene que brindar una pantalla donde se indique el Historial de Auditoria en la cual se listen todas las actividades realizadas en orden cronológico.</p> <p><b>CA6:</b> Debe de permitir la búsqueda por nombre de usuario y modulo y filtrado por fechas un día específico o en un rango de fechas.</p>

---

**CA7:** Los registros de auditoría no pueden ser editados ni eliminados por ningún usuario.

---

**CA8:** Para acceder a este apartado se debe de tener roles y permisos.

---

### 3.1.3 Product Backlog

Una vez establecidas las historias de usuario, el paso a seguir es estimarlas y priorizarlas en el Product Backlog, en una reunión en la que asistieron el Scrum Master, y el Product Owner [55].

A continuación, se presenta el Product Backlog en la Tabla Número 22.

*Tabla 27. Product Backlog*

<b>Orden</b>	<b>Código</b>	<b>Descripción</b>
1	HUM-01	Iniciar Sesión
2	HUM-02	Recuperar Contraseña
3	HUM-03	Gestionar mi perfil
4	HUM-04	Gestión de usuarios
5	HUM-05	Gestión roles y permisos
6	HUM-06	Gestión de Ubicaciones Geográficas
7	HUM-07	Gestión de Servicios
8	HUM-08	Gestión de Beneficios
9	HUM-09	Gestión de Socios
10	HUM-10	Buscar y Filtrar Socios
11	HUM-11	Control de Exonerados
12	HUM-12	Reporte de Socios
13	HUM-13	Gestión de Bloques
14	HUM-14	Gestión de Nichos

<b>15</b>	HUM-15	Ver Mapa Cementerio
<b>16</b>	HUM-16	Gestión de Fallecidos
<b>17</b>	HUM-17	Buscar Fallecidos
<b>18</b>	HUM-18	Inhumación (Asignar)
<b>19</b>	HUM-19	Exhumación
<b>20</b>	HUM-20	Reporte de Asignaciones
<b>21</b>	HUM-21	Ver deudas
<b>22</b>	HUM-22	Facturación
<b>23</b>	HUM-23	Registrar Pago
<b>24</b>	HUM-24	Auditoría (Logs)

## **3.2 Desarrollo Etapa Juego**

### **3.2.1 Sprint 0**

El objetivo en este Sprint 0 es definir las bases arquitectónicas que guiarán el desarrollo de la aplicación web para la gestión del Cementerio Indígena de Otavalo. A lo largo de esta fase se determinará la arquitectura general del proyecto. Tomando en cuenta todos los componentes relacionados en la aplicación.

Dentro de esto, se abarcará el diagrama de la base de datos, en la cual se detallan las entidades y relaciones, esenciales para la gestión del cementerio tanto en sus módulos como en finanzas, socios, fallecidos, nichos, etc.

Finalmente, se determinarán los métodos de seguridad y autenticación, basándose en el sistema nativo de Laravel para el control y monitoreo de sesiones, roles y permisos, lo que permitirá limitar el acceso a módulos sensibles como historial de pagos o asignaciones de fallecidos, asegurando la seguridad de los datos.

## 1. Definición de la Arquitectura General del Proyecto

La arquitectura general de la aplicación se encuentra representada en la Figura 6, la que explica el flujo de conexión entre los componentes de la aplicación web. Para el acceso a los servicios a cargo del personal administrativo de la organización UNORICO SAMASHUNCHIK se lleva a cabo mediante un navegador web, que actúa como cliente mandando peticiones HTTP de forma segura al servidor.

Para asegurar la seguridad, Laravel utiliza la capa de Middleware, esta verifica si un usuario tiene una sesión activa, tiene roles y permisos requeridos antes de tener acceso a la lógica del negocio. Si la verificación es satisfactoria, el controlador se comunica con los Modelos (Eloquent ORM) para obtener los datos de la base de PostgreSQL.

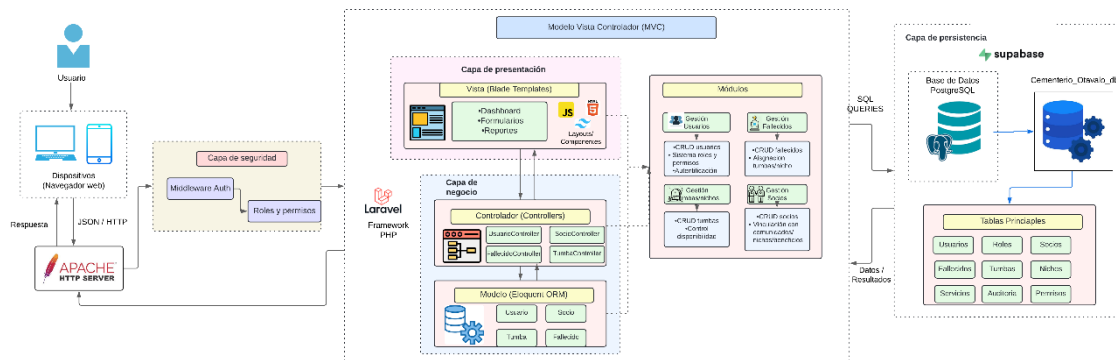


Figura 6 Arquitectura General del Proyecto

## 2. Definición de la estructura del software (Backend/Frontend)

La estructura de la aplicación web se basa en la estructura organizada del Framework Laravel que usa el patrón de diseño MVC (Model-View-Controller), el cual separa la lógica de negocio, la interfaz de usuario y los datos. Esto permite un desarrollo por componentes y mejora la mantenibilidad a largo plazo [56]. A continuación, se detalla la estructura de cada capa:

- **Frontend**

Esta es la encargada de la interfaz gráfica con la que interactúa el usuario final con la aplicación web. El objetivo que tiene es asegurar el uso y la adecuada visualización de datos.

- **Plantillas (Laravel Blade)**

El diseño de las vistas se realiza a través de **Blade**, lo cual son las plantillas nativas de Laravel. Esta permite la integración de la herencia de

plantillas y la definición de secciones, agilizando la reutilización de componentes clásicos como cabeceras, menús de navegación y pies de página en todas las pantallas de la aplicación web [57].

- **Tecnologías web (HTML5, CSS3, JavaScript)**

La arquitectura del contenido se lleva a cabo por HTML5, mientras que los estilos y la interacción por parte del cliente se gestiona por CSS3 y JavaScript, garantizando que la interfaz sea flexible y versátil para los diferentes dispositivos.

- **Backend (Lógica)**

Esta parte representa el núcleo de la aplicación, que se encarga de procesar las solicitudes HTTP, emplear las reglas de negocio de la organización y administrar la seguridad de la información.

- Framework Laravel

Este se encarga de gestionar el enrutamiento de URLs, validación de los datos de entrada y la respectiva autenticación de usuarios por medio de Middleware, el cual asegura que solo el personal autorizado con sus respectivos permisos y roles logre acceder a los módulos [58].

- ORM (Eloquent)

Para lograr la interacción con la base de datos se aplica **Eloquent ORM (Object-Relational Mapping)**. Este componente usa el patrón *Active Record*, posibilitando la interacción con las tablas de la base de datos PostgreSQL como fallecidos, socios, nichos, bloques, etc. Mediante modelos orientados a objetos en vez de consultas SQL complejas. En seguridad este se encarga de proteger automáticamente la aplicación contra ataques de inyección SQL por medio de sentencias preparadas [59].

- Base de datos PostgreSQL

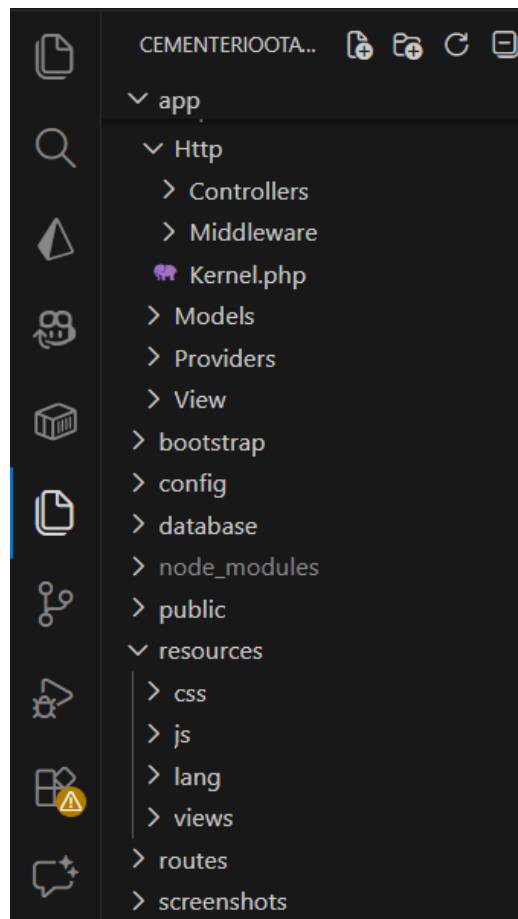
El módulo de gestión se apoya sobre una base de datos relacional PostgreSQL, seleccionada por su capacidad para manejar grandes volúmenes de datos y su extensión espacial PostGIS. Dado que la gestión actual del cementerio se realiza mediante registros físicos propensos al deterioro, se procedió a diseñar un modelo de datos desde cero que normaliza y digitaliza esta información [60].

El diseño contempla la creación de tablas específicas para cubrir los requerimientos de gestión de sepulturas, arrendamientos y ubicación geográfica.

### 3. Estructura general del proyecto

Este proyecto se estructura mediante directorios claves, los cuales dividen las responsabilidades lógicas de la aplicación tal como se indica en la figura 7:

1. **App/:** tiene toda la lógica central de la aplicación
  - a. **Models/:** Se determinan las entidades del negocio (Socio.php, Nicho.php, Fallecido.php) y las relaciones con la base de datos.
  - b. **Http/Controllers/:** Aloja los controladores que gestionan las solicitudes del usuario y gestionan las respuestas.
  - c. **Http/Middleware/:** Administra la seguridad (verificación de autenticación)
2. **Database:** Contiene las migraciones y seeders
  - a. **Migrations:** contiene el control de las versiones de la base de datos, facilitando crear y cambiar tablas.
3. **Resources/views/:** Contiene las plantillas visuales diseñadas con BLADE, lo cual permite generar interfaces dinámicas HTML.
4. **Routes/:** establece los puntos de entrada de la aplicación web y conexión con los controladores



*Figura 7 Estructura General del Proyecto*

4. **Diagrama Entidad–Relación.** La Figura 8 presenta el diagrama entidad-relación (ER) diseñado para la aplicación web.

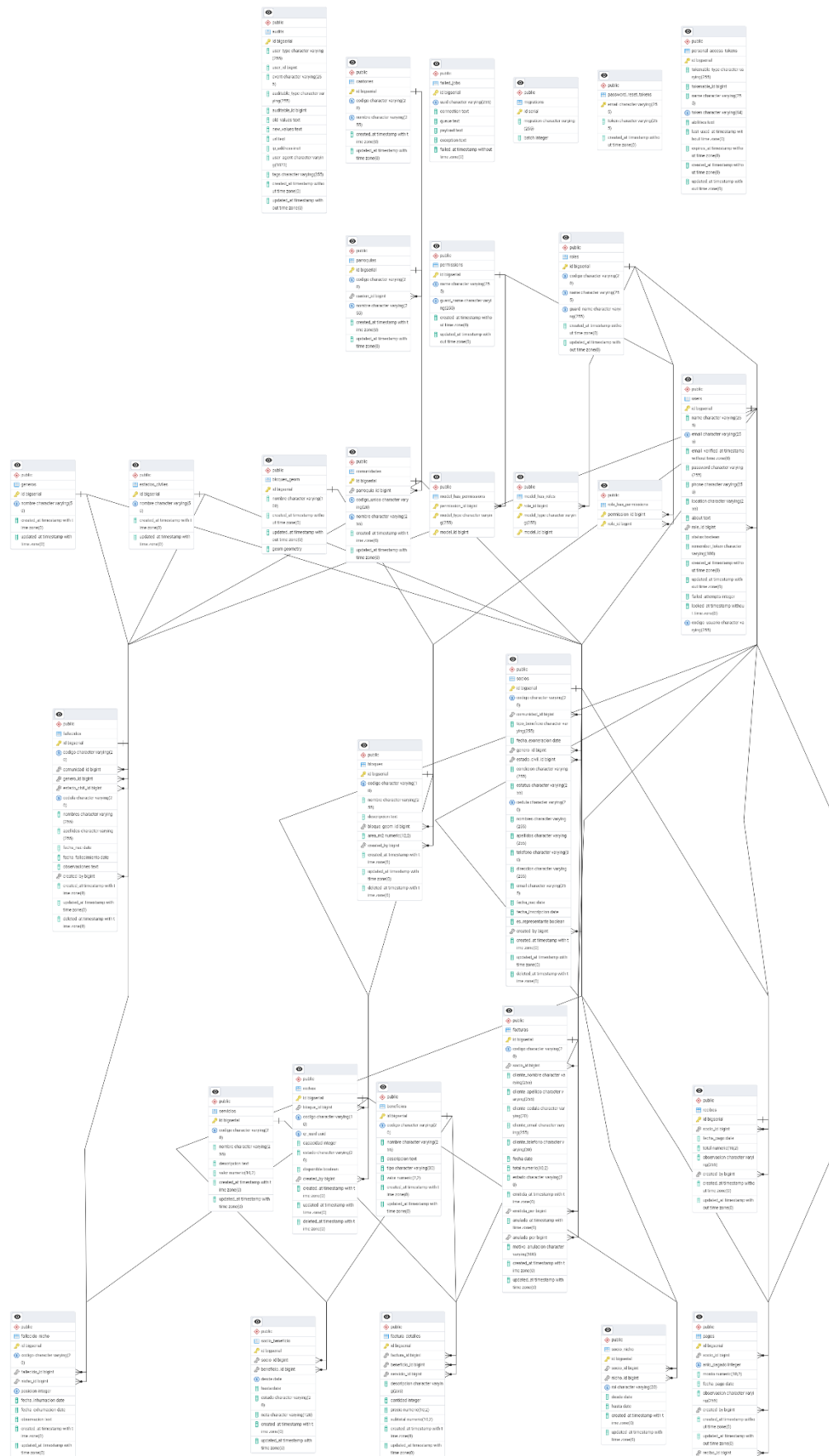


Figura 8 Diagrama Entidad Relación de la Base de Datos

## 5. Desglose de Entidades Principales

A continuación, se indican las entidades que conforman la base de la aplicación web con sus respectivos atributos más importantes, tipo de datos y uso dentro del aplicativo.

*Tabla 28. Tabla Socios*

<b>Socios</b>		
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del socio (PK).
codigo	varchar	Código interno administrativo del socio.
comunidad_id	bigint (int8)	(FK) Relación con la tabla comunidades.
tipo_beneficio	varchar	Categoría de subsidio (ej. "con subsidio", "sin_subsidio").
fecha_exoneracion	date	Fecha en la que aplicó una exoneración (si aplica).
genero_id	bigint (int8)	(FK) Relación con la tabla géneros.
estado_civil_id	bigint (int8)	(FK) Relación con la tabla estados_civiles.
condicion	varchar	Condición social o física (ej. "ninguna", "discapacidad").
estatus	varchar	Estado vital del socio (ej. "vivo", "fallecido").
cedula	varchar	Número de documento de identidad.
nombres	varchar	Nombres completos del socio.
apellidos	varchar	Apellidos completos del socio.
telefono	varchar	Número de teléfono de contacto.
direccion	varchar	Dirección domiciliaria.
email	varchar	Correo electrónico para notificaciones.
fecha_nac	date	Fecha de nacimiento.
fecha_inscripcion	date	Fecha en la que se registró en la organización.
created_by	bigint (int8)	(FK) ID del usuario administrador que creó el registro.
created_at	timestampz	Fecha y hora de creación del registro.
updated_at	timestampz	Fecha y hora de la última actualización.
deleted_at	timestampz	Fecha de eliminación lógica (Soft Delete).

Tabla 29. Tabla Fallecidos

<b>Fallecidos</b>		
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del fallecido (PK).
codigo	varchar	Código interno del expediente.
comunidad_id	bigint (int8)	(FK) Comunidad de procedencia.
genero_id	bigint (int8)	(FK) Género del fallecido.
estado_civil_id	bigint (int8)	(FK) Estado civil al momento del deceso.
cedula	varchar	Documento de identidad.
nombres	varchar	Nombres completos.
apellidos	varchar	Apellidos completos.
fecha_nac	date	Fecha de nacimiento.
fecha_fallecimiento	date	Fecha del deceso.
observaciones	text	Notas adicionales sobre el deceso o traslado.
created_by	bigint (int8)	Usuario que registró al fallecido.
created_at	timestampz	Fecha de creación.
updated_at	timestampz	Fecha de actualización.
deleted_at	timestampz	Fecha de eliminación lógica.

Tabla 30. Tabla Bloques

<b>Bloques</b>		
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del bloque (PK).
codigo	varchar	Código administrativo único (ej. "B-01").
nombre	varchar	Nombre común del bloque (ej. "San Pedro").
descripcion	text	Detalles adicionales sobre el terreno o ubicación.
bloque_geom_id	bigint (int8)	(FK) Referencia a la tabla geométrica del mapa.
area_m2	numeric	Área total del bloque en metros cuadrados.
created_by	bigint (int8)	(FK) Usuario que creó el registro.
created_at	timestampz	Fecha de creación del registro.
updated_at	timestampz	Fecha de la última actualización.

deleted_at	timestampz	Fecha de eliminación lógica (Soft Delete).
------------	------------	--

*Tabla 31. Tabla Nichos*

<b>Nichos</b>		
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del nicho (PK).
socio_id	bigint (int8)	(FK) Socio responsable actual (si está asignado).
bloque_id	bigint (int8)	(FK) Bloque físico al que pertenece.
nicho_geom_id	bigint (int8)	(FK) Enlace con la geometría del mapa.
codigo	varchar	Código visual único (ej. B1-N5).
qr_uuid	uuid	Identificador único para generación de código QR.
tipo_nicho	varchar	Categoría de propiedad (ej. "PROPIO", "ALQUILADO").
clase_nicho	varchar	Tipo físico (ej. "BOVEDA", "SUELO").
capacidad	smallint (int2)	Número máximo de cuerpos (Default: 3).
ocupacion	smallint (int2)	Cantidad actual de ocupantes.
disponible	boolean (bool)	Estado de disponibilidad (True=Libre, False=Ocupado).
estado	varchar	Condición de mantenimiento (ej. "BUENO").
descripcion	text	Observaciones adicionales.
created_by	bigint (int8)	Usuario que creó el registro.
created_at	timestampz	Fecha de creación.
updated_at	timestampz	Fecha de actualización.
deleted_at	timestampz	Fecha de eliminación lógica.

*Tabla 32. Tabla Fallecido\_Nicho*

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del registro (PK).
codigo	varchar	Código interno del expediente de inhumación.
fallecido_id	bigint (int8)	(FK) Identificador de la persona fallecida.
socio_id	bigint (int8)	(FK) Socio que autorizó o gestionó el trámite.
nicho_id	bigint (int8)	(FK) Nicho donde se realiza la inhumación.

posicion	integer (int4)	Ubicación dentro del nicho (ej. 1, 2, 3).
fecha_inhumacion	date	Fecha en la que se realizó el entierro.
fecha_exhumacion	date	Fecha de exhumación (si aplica).
observacion	text	Notas adicionales sobre el procedimiento.
created_at	timestampz	Fecha y hora de registro.
updated_at	timestampz	Fecha y hora de la última actualización.

*Tabla 33. Tabla Pagos*

<b>Pagos</b>		
<b>Campo</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del pago (PK).
socio_id	bigint (int8)	(FK) Socio que realiza el aporte.
recibo_id	bigint (int8)	(FK) Número de recibo físico o digital asociado.
anio_pagado	integer (int4)	Año fiscal o periodo que cubre el pago.
monto	numeric	Cantidad de dinero cancelada.
fecha_pago	date	Fecha en la que se realizó la transacción.
observacion	varchar	Notas adicionales sobre el pago.
created_by	bigint (int8)	(FK) Usuario administrativo que registró el cobro.
created_at	timestamp	Fecha de creación del registro.
updated_at	timestamp	Fecha de la última modificación.

*Tabla 34. Usuarios*

<b>Usuarios</b>		
<b>Columna</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
id	bigint (int8)	Identificador único del usuario (PK).
codigo_usuario	varchar	Código interno administrativo.
role_id	bigint (int8)	(FK) Nivel de acceso (ej. Administrador, secretaria).
name	varchar	Nombre completo del usuario.
email	varchar	Correo electrónico (usado para Login).
password	varchar	Contraseña encriptada (Hash).
phone	varchar	Teléfono de contacto.

location	varchar	Ubicación o dirección del usuario.
about	text	Breve descripción biográfica.
status	boolean (bool)	Estado de la cuenta (True=Activa, False=Bloqueada).
failed_attempts	integer (int4)	Contador de intentos fallidos de login (Seguridad).
locked_at	timestamp	Fecha de bloqueo por seguridad.
email_verified_at	timestamp	Fecha de verificación del correo.
remember_token	varchar	Token para "recordar sesión".
created_at	timestamp	Fecha de creación del usuario.
updated_at	timestamp	Fecha de última modificación.

### 3.2.2 Sprint 1

- Planificación – Sprint 1

El objetivo principal de este primer sprint es realizar el levantamiento del proyecto, configurar el entorno de trabajo y garantizar que solo personal autorizado pueda acceder a la aplicación. En este sprint se abarcarán las historias de usuario desde la HUM-01 hasta la HUM-06

*Tabla 35. Sprint Backlog – Sprint 1*

Historia / Código	Nombre	Tarea
CONF-01	Inicialización de Proyecto	<b>Entorno:</b> Instalar Laravel y configurar la ejecución con el comando <code>php artisan serve</code> . <b>Git:</b> Inicializar repositorio, crear archivo <code>.gitignore</code> y subir primer commit a ramas main. Luego crearse una rama secundaria llamada develop para empezar el desarrollo
CONF-02	Conexión Base de Datos	<b>PostgreSQL Local (pgAdmin):</b> Crear una base de datos y configurar las credenciales para él <code>.env</code> de Laravel.

		<p><b>Supabase (Web):</b> Crear proyecto en Supabase y obtener credenciales de la base de datos PostgreSQL.</p> <p><b>Laravel:</b> Configurar archivo .env (DB_HOST, DB_PORT, DB_USERNAME) y probar conexión.</p>
<b>CONF-03</b>	Instalación Dependencias	<p><b>Composer:</b> Instalar paquetes base: spatie/laravel-permission, barryvdh/laravel-dompdf, maatwebsite/excel.</p> <p><b>Frontend:</b> Instalar Bootstrap/Tailwind y configurar compilación con Vite.</p>
<b>HUM-01</b>	Iniciar Sesión	<p><b>Backend:</b> Implementar LoginController y validación de credenciales encriptadas.</p> <p><b>Frontend:</b> Diseñar vista Blade auth.login con formulario responsivo y manejo de errores.</p>
<b>HUM-02</b>	Recuperar Contraseña	<p><b>Backend:</b> Configurar servicio de correo (SMTP) en el .env para envío de tokens de recuperación.</p> <p><b>Frontend:</b> Crear vistas para solicitud de enlace y formulario de nueva contraseña.</p>
<b>HUM-03</b>	Gestionar mi perfil	<p><b>Backend:</b> Desarrollo de lógica para actualizar datos personales como: nombres, apellidos, teléfono, correo, etc. del usuario autenticado</p> <p><b>Frontend:</b> Vista de perfil de usuario para cambios en los datos personales.</p>
<b>HUM-04</b>	Gestión de Usuarios	<p><b>Base de Datos:</b> Crear Seeder para tres tipos de usuario “Administrador, Usuario y Auditor”</p> <p><b>Backend:</b> CRUD de usuarios (Crear/Editar/Eliminar) exclusivo para el rol Administrador.</p>
<b>HUM-05</b>	Roles y Permisos	<p><b>Backend:</b> Configurar Middleware de Spatie para restringir rutas según el rol que se le dé y los permisos que se concedan “Administrador”</p> <p><b>Base de Datos:</b> Definir roles iniciales mediante un Seeder de permisos y roles para que se creen directamente en la base.</p>

<b>HUM-06</b>	Ubicaciones Geográficas	<p><b>Base de Datos:</b> Migraciones para cantones, parroquias y comunidades u organizaciones.</p> <p><b>Backend:</b> Controlador cuando selecciono un cantón me despliega que parroquias hay y selecciono una de ellas para ver que comunidades hay para seleccionar</p>
---------------	----------------------------	---

Durante este Sprint 1 se configuran las bases para inicializar el proyecto como: subir el proyecto en Github, instalar dependencias necesarias, configuración del archivo .env para la base de datos y algunos controles como cierre de sesión, correos smtp entre otros. Una vez configurado todo, comenzamos con las historias de usuario.

- Incremento – Sprint 1

En la HUM-01 (Iniciar Sesión), relacionada con la Tabla 3, se comienza con el desarrollo del LoginController en el backend para validar las credenciales encriptadas de los usuarios de la aplicación. Para comprobar el correcto funcionamiento se crearon Seeders que agregarán usuarios de prueba a la base. Al ingresar a la ruta principal, este se redirige a la vista Login, donde se nos indica un formulario responsivo donde se ingresan las credenciales, si son correctas, permite el ingreso al dashboard en caso de ser incorrectas, nos indica los mensajes de error de validación. Esto se ilustra en las Figuras 9, 10, 11.

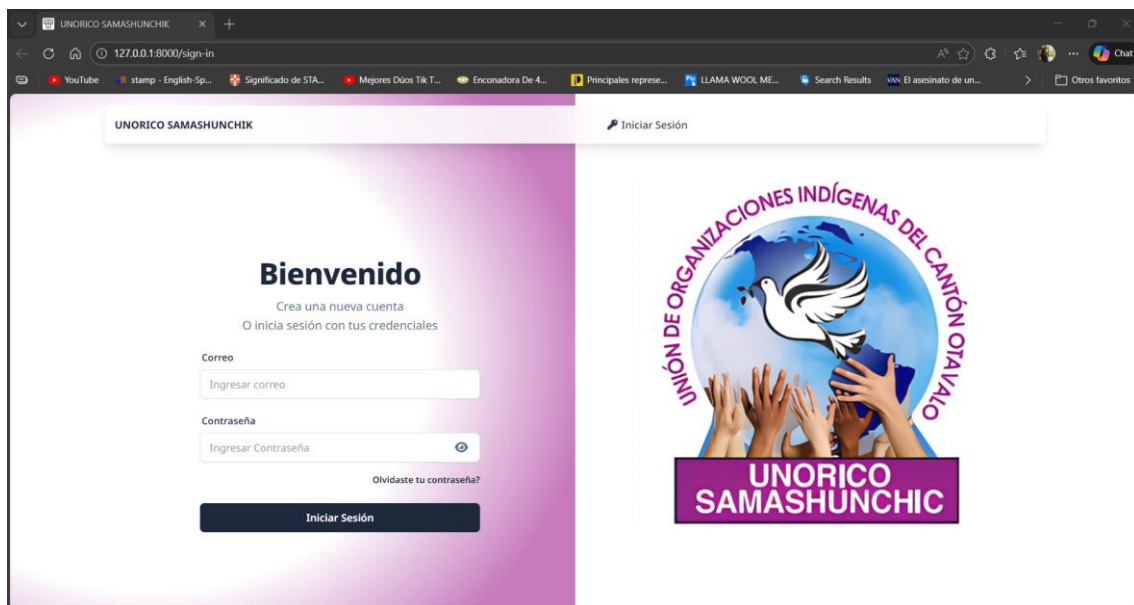
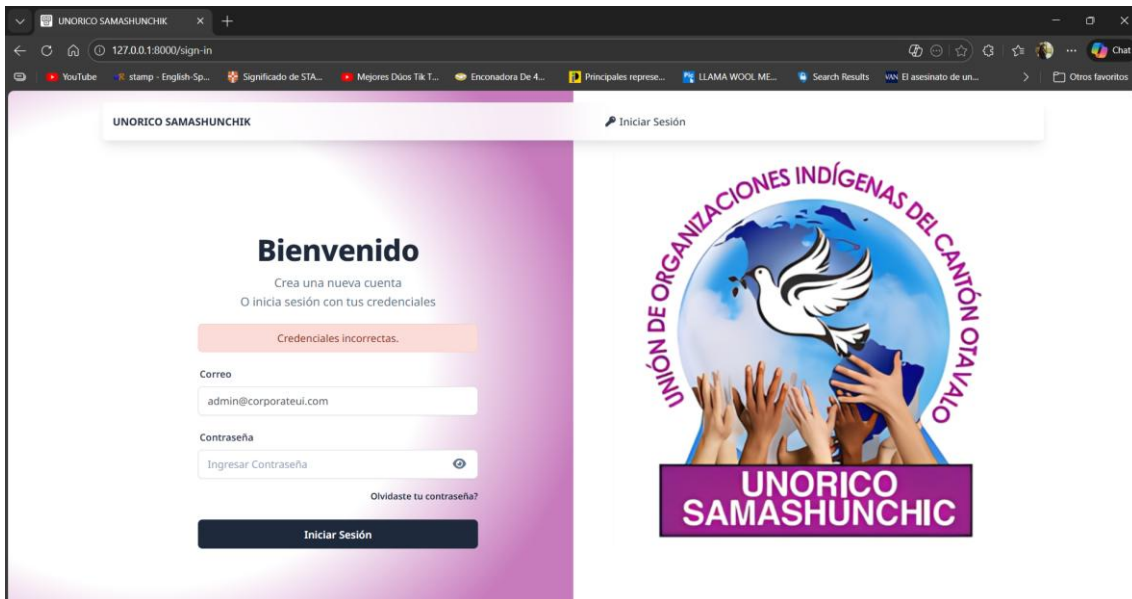
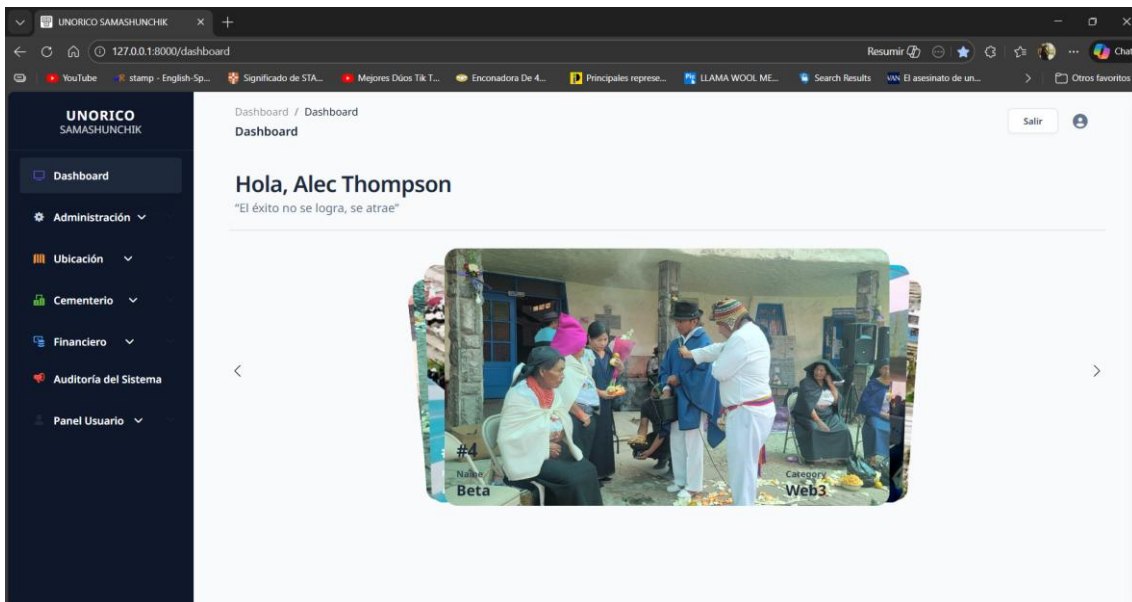


Figura 9 Iniciar Sesión

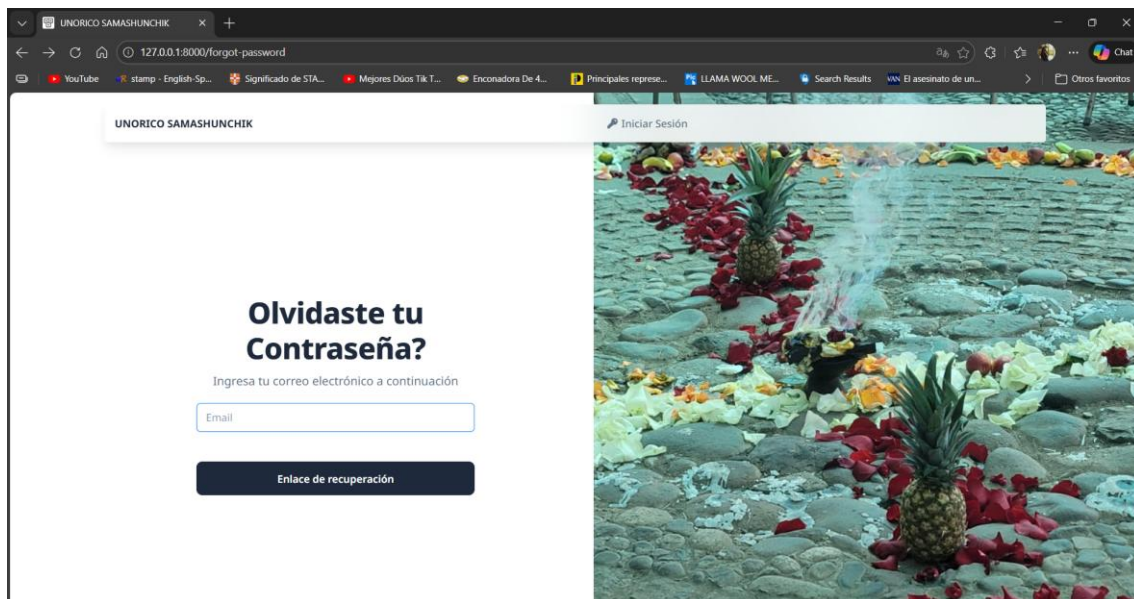


*Figura 10 Credenciales Incorrectas*



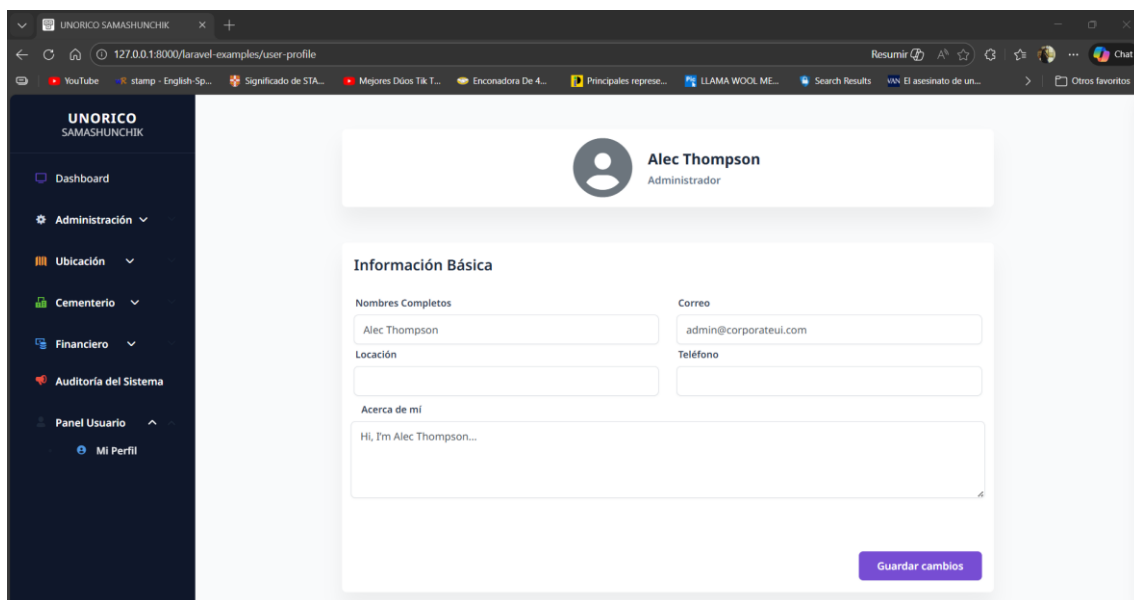
*Figura 11 Dashboard (Ingreso Exitoso)*

A continuación, se empezó con el desarrollo de la HUM-02 (Recuperar Contraseña), corresponde a la Tabla 4, se inició en la configuración del servicio de correo (SMTP) en el archivo .env con el objetivo de permitir el envío de tokens de recuperación. Esto se probó desde la pantalla de Login. Al pulsar la opción de “olvidaste tu contraseña”, la aplicación se dirige a la vista de la solicitud de enlace en la cual el usuario debería de ingresar su correo para que le llegue un correo donde podrá ingresar su nueva contraseña. Esto se ilustra en la Figura 12.



*Figura 12 ¿Olvidaste tu contraseña?*

Posteriormente se procedió a desarrollar la HUM-03 (Gestionar mi perfil), relacionada con la Tabla 5, se empezó con el desarrollo del backend para actualizar los datos del usuario verificado como, por ejemplo, nombres y apellidos, teléfono, dirección, correo y descripción. Se probó este apartado cambiando la información de una cuenta activa. Al entrar a la sección de “Mi perfil”, se nos muestra una vista con un formulario con los datos actuales. En la figura 13 se muestra el apartado.



*Figura 13 Mi perfil*

Seguidamente, se realizó la HUM-04 (Gestión de Usuarios), la cual corresponde a la Tabla 6, se inició con la creación de un Seeder para ingresar a la base 3 tipos de usuario como. Administrador, Usuario y Auditor.

Luego se desarrolló el CRUD completo que tiene las acciones: crear, editar/actualizar y eliminar. Al acceder al módulo de gestión de usuarios, se muestra una tabla con los registros con sus respectivas funciones. Como se ilustra en la figura 14.

#	CÓDIGO	NOMBRE	EMAIL	UBICACIÓN	ROL	ESTADO	ACCIONES
1	US0001	Alec Thompson Sin teléfono	admin@corporateui.com	N/A	Administrador	Activo	[Edit]
2	US0002	María López Sin teléfono	usuario@corporateui.com	N/A	Usuario	Inactivo	[Edit]
3	US0003	Carlos Pérez Sin teléfono	auditor@corporateui.com	N/A	Auditor	Inactivo	[Edit]

*Figura 14 Gestión Usuarios*

Después se llevó a cabo la HUM-05 (Gestión de Roles y Permisos) relacionada con la Tabla 7, se empezó definiendo los permisos y roles que se usarán por medio de un Seeder para que estos ya se encuentren en la base de datos. En el backend, se configuró el Middleware del paquete Spatie para resguardar las rutas de la aplicación. En los apartados de roles y permisos se tiene el CRUD completo de roles y en permisos se seleccionan los permisos que se desea dar a un rol específico. Las figuras 15 y 16 indican estos apartados.

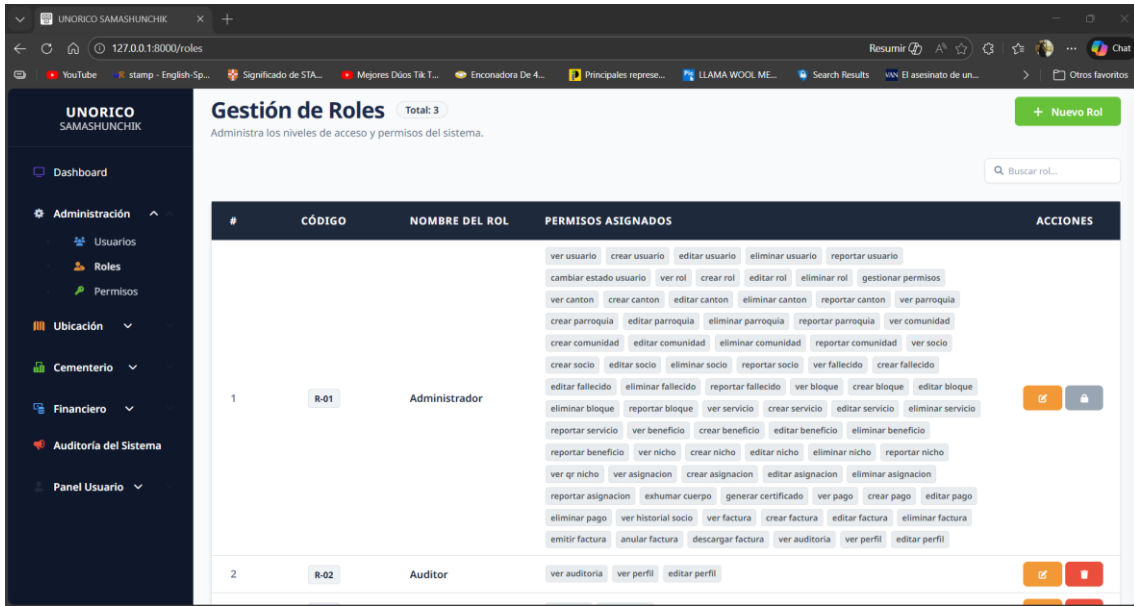


Figura 15 Gestión Roles

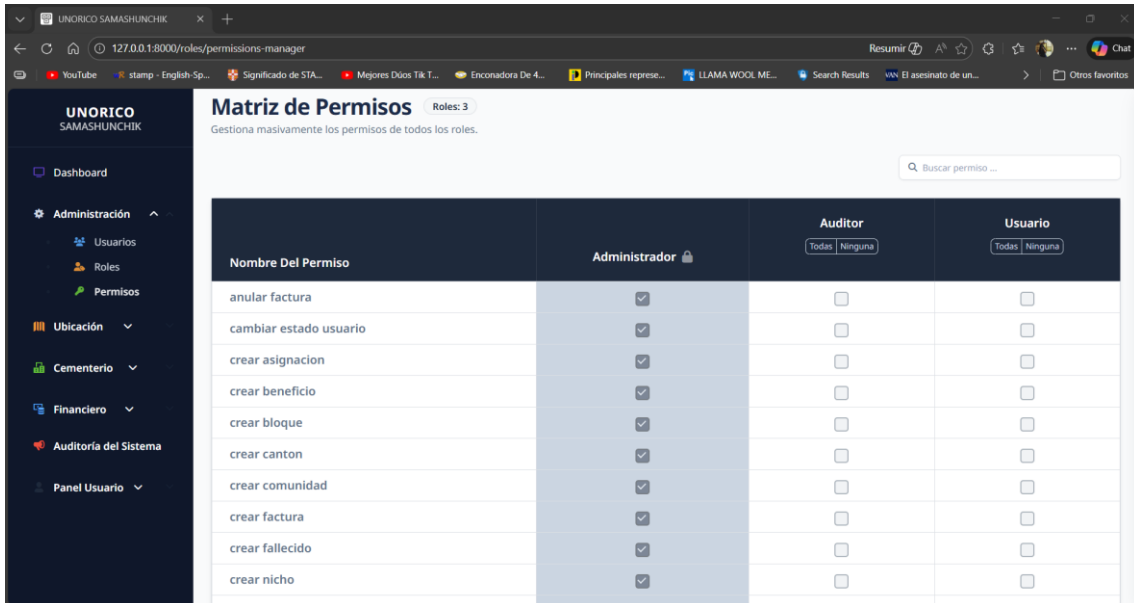
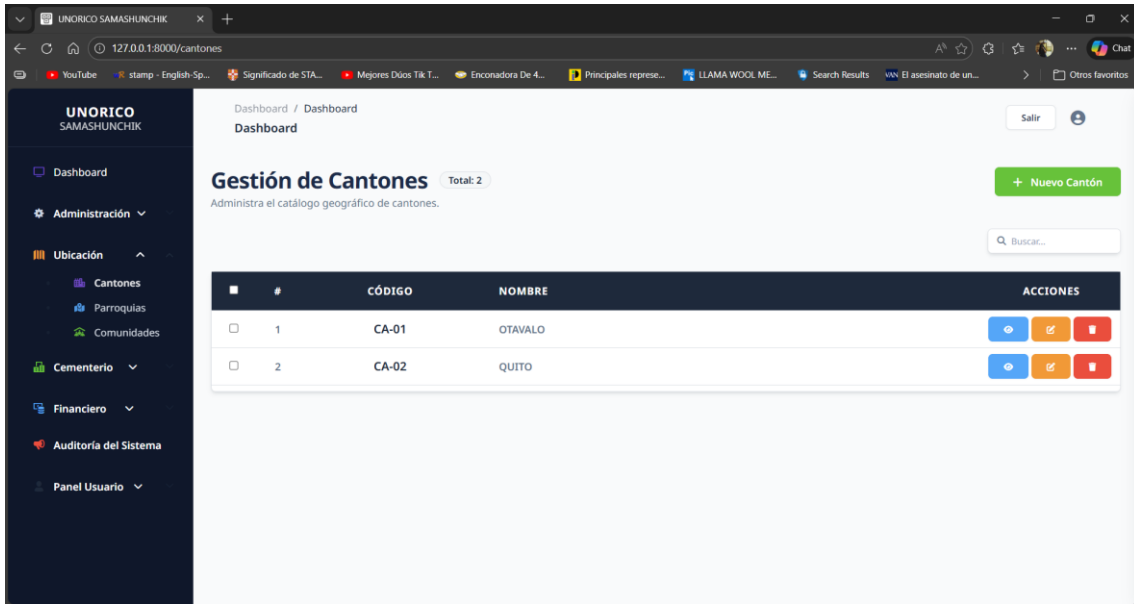
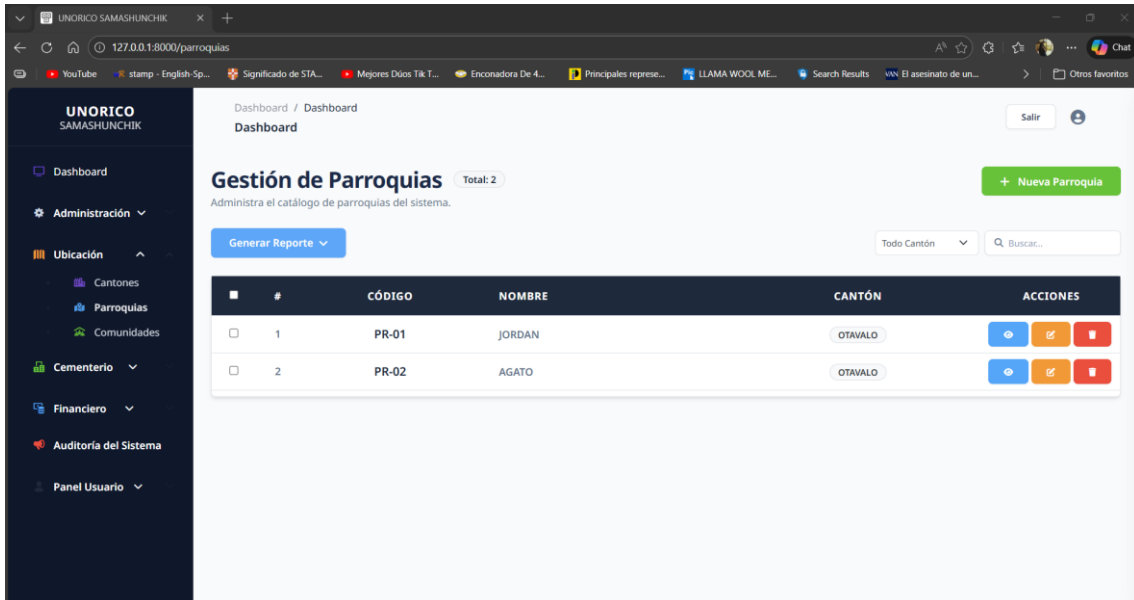


Figura 16 Gestión Permisos

Para cerrar el Sprint 1 se desarrolló la HUM-06 (Ubicaciones Geográficas) correspondiente a la Tabla 8, primero se creó la lógica del backend para cada apartado que conforma la gestión de ubicación que son: cantón, parroquia y comunidad u organización. En la interfaz de cada uno se tienen CRUDs completos con sus respectivas funciones. Esto se ilustra en las figuras 17, 18 y 19.



*Figura 17 Gestión de Cantones*



*Figura 18 Gestión de Parroquias*

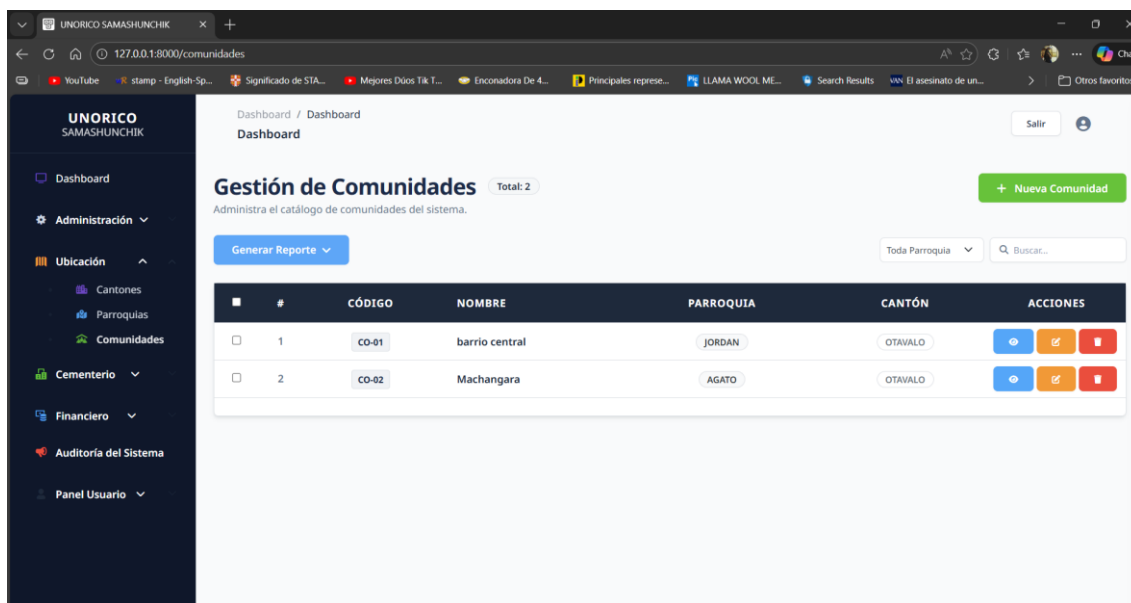


Figura 19 Gestión de comunidades u organizaciones

- Revisión – Sprint 1

A continuación, se explica el cumplimiento de los criterios de aceptación de las Historias de Usuario que fueron elaboradas en el primer Sprint. Se pueden apreciar estos criterios en la Tabla 36.

Tabla 36 Revisión - Criterios de Aceptación Sprint 1

Historias de Usuario	Criterio de Aceptación	Cumple
<b>Iniciar Sesión</b>	CA1: La aplicación debe de mostrar una pantalla de inicio de sesión que contenga los campos correo y contraseña.	SI
	CA2: Los campos correo y contraseña deben de ser obligatorios. En caso de que alguno se encuentre vacío debe de indicarse un mensaje de error.	SI
	CA3: El campo de correo debe de verificar que sea un email válido y debe de indicar un mensaje mostrando de cómo debería de ser un correo correcto.	SI
	CA4: Si el correo ingresado no existe debe de salir un mensaje “Usuario no existe”.	SI
	CA5: Si la contraseña ingresada no es correcta debe de salir un mensaje “Credenciales Incorrectas”.	SI
	CA6: Si al iniciar sesión es exitoso, el usuario es redirigido al panel del “Dashboard”.	SI
	CA7: El campo contraseña debe de tener el icono del ojo que permite visualizar y ocultar la contraseña.	SI

	CA8: Si un usuario intenta ingresar a alguna ruta sin autenticación será redirigido al login.	SI
	CA9: La aplicación permite cerrar sesión en cualquier instante y se redirige a la página de login.	SI
<b>Recuperar Contraseña</b>	CA1: La pantalla de login debe de mostrar un enlace “Olvidaste tu contraseña?” que me redirija a la vista o página de recuperar contraseña.	SI
	CA2: La página de recuperar contraseña debe de tener un campo donde pueda ingresar el correo de manera obligatoria y diga de título “¿Olvidaste tu contraseña?”.	SI
	CA3: El campo del correo debe de validar que sea un correo valido. Si es un correo correcto el borde de este cambia de color verde en caso contrario le indica con un mensaje como es un correo válido.	SI
	CA4: Cuando se envía el formulario, la aplicación debe de enviar un enlace de restablecimiento al correo registrado	SI
	CA5: Si el correo no se encuentra registrado la aplicación debe de mostrar un mensaje de “No encontramos un usuario con ese correo”	SI
	CA6: El enlace que fue enviado al correo debe de redirigir a la vista para restablecer la contraseña.	SI
	CA7: La nueva contraseña debe de cumplir con ciertos requisitos de seguridad: <ul style="list-style-type: none"> <li>• Mínimo 8 caracteres</li> <li>• Al menos una letra mayúscula</li> <li>• Al menos una letra minúscula</li> <li>• Al menos un número</li> <li>• Al menos un carácter especial (!@#\$...)</li> </ul>	SI
	CA8: Los campos para la nueva contraseña deben de tener el icono para visualizar y ocultar la contraseña.	SI
	CA9: Se deben de confirmar que las contraseñas coincidan en caso de no hacerlo mostrar un mensaje “Las contraseñas no coinciden”.	SI
	CA10: Al restablecer la contraseña el contador de intentos fallidos vuelve a 0 y en caso de estar bloqueada la cuenta se desbloquea.	SI
	CA11: Cuando completa el restablecimiento de la contraseña es redirigido de vuelta a la página de Login.	SI
<b>Gestionar Mi Perfil</b>	CA1: La aplicación debe de tener un apartado “Mi perfil” ingresando se muestre la información del usuario como rol y nombre.	SI
	CA2: La página del perfil, debe de contar con un formulario que contiene los campos : Nombres completos, correo, locación, teléfono y un acerca de mí con datos anteriormente llenados.	SI

	CA3: Los campos nombres completos, correo son obligatorios mientras que los demás campos son opcionales.	SI
	CA4: El correo debe de ser único en la aplicación. En caso de que el usuario lo cambie por uno ya registrado le saldrá un mensaje de error.	SI
	CA5: Cuando se guarden los cambios realizados exitosamente, se debe de mostrar un mensaje “perfil actualizado correctamente”.	SI
	CA6: Cuando intente guardar los cambios realizados y ocurre un error debe de salir un mensaje “Error al actualizar perfil”.	SI
	CA7: Los mensajes de éxito y error deben solo de durar 3 segundos y desaparecer automáticamente una vez culminado el tiempo estimado.	SI
<b>Gestión de usuarios</b>	CA1: La aplicación debe de permitir crear un nuevo usuario desde una ventana (modal) con los campos: Nombres completos, correo, contraseña, telefono, ubicación, rol y estado que se le otorga al usuario.	SI
	CA2: Los campos Nombres completos, Correo, Contraseña y Rol son obligatorios.	SI
	CA3: El correo electrónico debe de ser único. En caso de que ya se encuentre registrado debe de indicar “Este correo ya está registrado”.	SI
	CA4: El campo de correo debe de validar el formato de email valido. Indicando un mensaje de cómo es un correo correcto.	SI
	CA5: La contraseña debe de contar con ciertos requisitos como: <ul style="list-style-type: none"> <li>• Mínimo 8 caracteres</li> <li>• Al menos una letra mayúscula</li> <li>• Al menos una letra minúscula</li> <li>• Al menos un número</li> <li>• Al menos un carácter especial (@\$!%*?&amp;)</li> </ul>	SI
	CA6: El campo de contraseña debe de contar con el icono del ojo para mostrar u ocultar la contraseña.	SI
	CA7: El campo de teléfono debe de solo aceptar números con un máximo de 10 dígitos.	SI
	CA8: El código del Usuario se debe de generar automáticamente cuando se crea uno nuevo.	SI
	CA9: Cuando se crea un nuevo usuario exitosamente se debe de indicar un mensaje “El usuario xxxx fue creado exitosamente”.	SI

	CA10: La aplicación debe permitir editar los datos de un usuario registrado.	SI
	CA11: Cuando se edite el correo debe de seguir siendo único en cuanto a los demás usuarios.	SI
	CA12: Una vez actualizados los datos se muestra un mensaje “El usuario xxxx fue actualizado correctamente”.	SI
	CA13: Es necesario permitir el activar o desactivar una la cuenta de un usuario mediante un toggle. Para no eliminarlo y no perder su historial de actividades.	SI
	CA14: La página principal debe de indicar un listado de los usuarios existentes en una tabla con los siguientes datos: Nombres, correo, rol y estado con su respectiva paginación.	SI
	CA15: El buscador tiene que permitir filtrar usuarios por nombres, correo o su código único.	SI
	CA16: La aplicación debe generar reportes de todos los usuarios o seleccionar algunos en formato PDF e Excel con todos los campos necesarios: Nombres, correo, teléfono, ubicación, rol y su estado.	SI
	CA17: Si no se selecciona ningún usuario cuando se intente generar el reporte mostrar un mensaje “Debe de seleccionar al menos un usuario”.	SI
<b>Gestión de Roles y Permisos</b>	CA1: El sistema debe permitir crear un nuevo Rol (ej. "Secretario") desde una ventana (modal) que tenga el campo nombre del rol y debe de ser obligatorio.	SI
	CA2: El nombre del rol debe de ser único caso contrario mostrar un mensaje de error.	SI
	CA3: El código único se genera automáticamente cuando se genera un nuevo registro.	SI
	CA4: Al crear el rol exitosamente indicar un mensaje “El rol xxxx ha sido creado correctamente”.	SI
	CA5: La aplicación tiene que permitir editar el nombre de un rol existente desde un modal.	SI
	CA6: No debe permitirse renombrar el rol “Administrador”. Si trata de hacerlo mostrar “No se puede renombrar el rol Administrador”.	SI
	CA7: Al actualizar el nombre de un rol exitosamente indicar “Rol xxxx actualizado correctamente”.	SI
	CA8: La aplicación debe de permitir eliminar un rol mediante una confirmación previa.	SI
	CA9: No se puede eliminar el rol “Administrador”. Si hace el intento mostrar “No se puede eliminar, existe usuarios utilizando este rol”.	SI

	CA10: Al crear el rol exitosamente indicar un mensaje “El rol xxxx ha sido creado correctamente”.	SI
	CA11: Cuando se eliminar un rol exitosamente presentar un mensaje “El rol xxxx ha sido eliminado correctamente”.	SI
	CA12: La aplicación debe de indicar un apartado de permisos para cada rol existente.	SI
	CA13: Cada uno de los permisos debe de mostrarse con un checkbox el cual se puede marcar o desmarcar individualmente.	SI
	CA14: Cada grupo de permisos para cada rol debe de tener los botones “Todos” o “Ninguno” para seleccionar o deseleccionar todos los permisos en masa.	SI
	CA15: Una vez guardado los cambios se debe de mostrar un mensaje “Los permisos han sido actualizados correctamente”.	SI
	CA16: El listado en la tabla de roles debe de indicar: código, nombre y los permisos asignados que tiene cada rol con su respectiva paginación.	SI
<b>Gestión de Ubicaciones Geográficas</b>	CA1: La aplicación tiene que permitir crear, editar eliminar y buscar cantones. El campo nombre debe ser obligatorio y se convierte automáticamente en mayúsculas.	SI
	CA2: El nombre el cantón debe ser único. Si se reitera mostrar “Este nombre del cantón ya está registrado”.	SI
	CA3: El código del cantón se genera automáticamente cuando se registra o crea uno nuevo.	SI
	CA4: No es posible eliminar un cantón que tenga parroquias vinculadas. En caso de hacerlo indicar “No se puede eliminar tiene parroquias asociadas”.	SI
	CA5: El buscar debe de filtrar por nombre o código del Cantón.	SI
	CA6: La aplicación debe permitir crear, editar, eliminar y buscar parroquias. El campo nombre tiene que ser obligatorio y convertido en mayúsculas.	SI
	CA7: El nombre de la parroquia debe ser único dentro del mismo cantón. Si se repite presentar “Ya existe una parroquia con ese nombre en el cantón seleccionado”.	SI
	CA8: El código de la parroquia se genera automáticamente.	SI
	CA9: La tabla donde está el listado de las parroquias tiene que filtrar por cantón por medio un select desplegable.	SI
	CA10: El buscar debe de filtrar tanto por nombre como por el código único de la parroquia.	SI

CA11: Se tiene que generar reportes en formato PDF e Excel de todas las parroquias o solo de las seleccionadas.	SI
CA12: La aplicación debe permitir crear, editar, eliminar y buscar comunidades u organizaciones. El campo nombre tiene que ser obligatorio.	SI
CA13: El nombre de la comunidad tiene que ser único dentro de la misma parroquia. En caso de repetirse mostrar “La comunidad xxxx ya existe en la parroquia xxxx”.	SI
CA14: El código único de la comunidad se genera automáticamente.	SI
CA15: La tabla donde se encuentra el listado de registro debe permitir el filtrado por parroquias a través de un select desplegable.	SI
CA16: Se debe de generar reportes en formato PDF y Excel de todas las comunidades o de ciertas seleccionadas.	SI
CA17: Cuando se seleccione la ubicación en los otros formularios como socio u otros las parroquias tienen que cargarse dinámicamente. Lo mismo sucede cuando selecciono una parroquia las comunidades se cargan automáticamente.	SI

- Retrospectiva – Sprint 1

En la reunión retrospectiva del Sprint 1, se comentó que el más grande reto fue establecer la arquitectura de seguridad del proyecto, implementando autenticación por medio de roles y permisos configurada mediante Spatie (RBCA) para asegurar que el acceso a los datos sensibles sea controlado. Igualmente, la incorporación de operaciones críticas de autogestión como recuperar la contraseña por medio de tokens seguros y actualización de perfiles mejora la eficiencia interna, creando bases sólidas en la aplicación perfectamente preparadas para funcionar con seguridad absoluta desde principio a fin.

### 3.2.3 Sprint 2

- Planificación – Sprint 2

En este segundo sprint se desarrollaron los apartados que son servicios y beneficios que brinda la organización, igualmente se priorizó el desarrollo del módulo de gestión de socios, el cual es muy esencial para la gestión entera del cementerio por parte de la administración. En este sprint se abarcarán las historias de usuario desde la HUM-07 a la HUM-12

Tabla 37. Sprint Backlog – Sprint 2

Historia / Código	Nombre	Tarea
HUM-07	Gestión de Servicios	<p><b>Backend:</b> CRUD de los Servicios que brinda la asociación y que se usarán en facturación.</p> <p><b>Frontend:</b> Tabla administrativa para editar precios, nombres de los servicios "Mantenimiento", "Uso Capilla", etc.</p>
HUM-08	Gestión de Beneficios	<p><b>Backend:</b> CRUD de los Beneficios que tiene la asociación y que se usarán en facturación.</p> <p><b>Frontend:</b> Tabla administrativa para editar precios, nombres de los beneficios "Inscripción", "Exonerado", etc.</p>
HUM-09	Gestión de Socios	<p><b>Backend:</b> Controlador SocioController con sus respectivas funciones para la relación con otras tablas.</p> <p><b>Frontend:</b> Formulario de registro completo con selectores anidados (Cantón -&gt; Parroquia -&gt; Comunidad).</p>
HUM-10	Buscar Socios	<p><b>Backend:</b> Implementar Query Scopes para filtrar por Cédula o Apellidos en tiempo real.</p> <p><b>Frontend:</b> Buscador AJAX o Datatable para filtrar el padrón sin recargar página.</p>
HUM-11	Control Exonerados	<p><b>Backend:</b> Lógica para vincular un socio con un beneficio dentro de la misma tabla socios y verificar edad para ser exonerado</p> <p><b>Frontend:</b> Dropdown que habilita el selector de beneficios en el formulario de socio.</p>

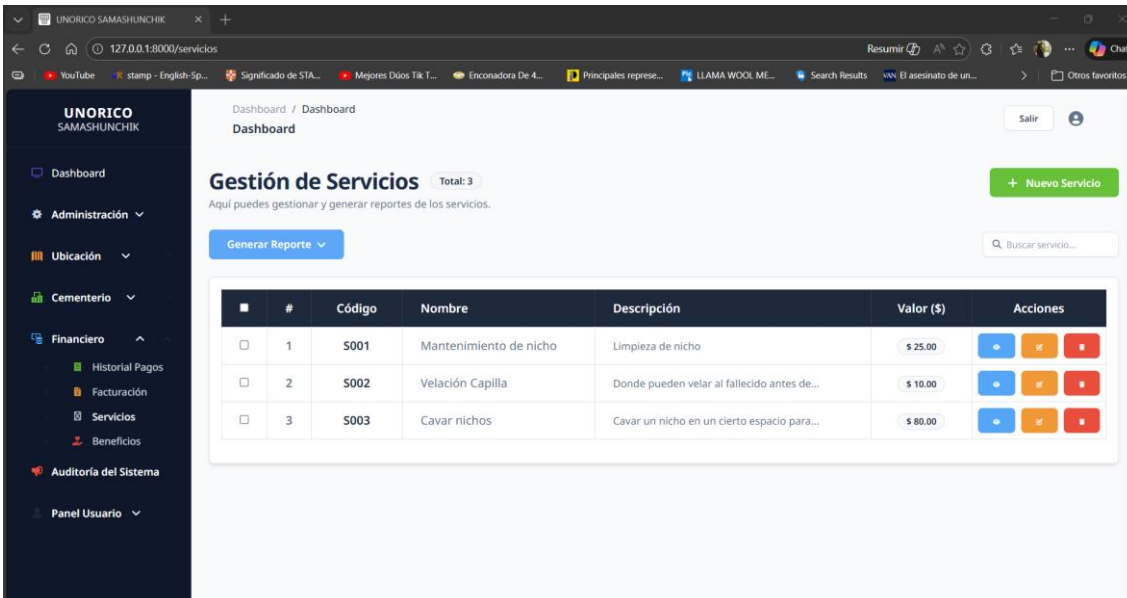
---

**HUM-12**      Reporte de Socios      **Backend:** Generar clase SociosExport para Excel y vista Blade para PDF.

**Frontend:** Botones de descarga "Generar Reporte" en la vista principal de socios.

---

Para la HUM-07 (Gestión de Servicios) relacionada a la Tabla 9, se enfocó en realizar un CRUD el cual está destinado para la administración del catálogo con sus respectivos precios de los servicios que se brindan y que se usarán en facturación. En la vista Servicios se puede apreciar algunos de los servicios que existen. Esto se ilustra en la Figura 20.



#	Código	Nombre	Descripción	Valor (\$)	Acciones
1	S001	Mantenimiento de nicho	Limpieza de nicho	\$ 25.00	[Iconos de acción]
2	S002	Velación Capilla	Donde pueden velar al fallecido antes de...	\$ 10.00	[Iconos de acción]
3	S003	Cavar nichos	Cavar un nicho en un cierto espacio para...	\$ 80.00	[Iconos de acción]

*Figura 20 Gestión de Servicios*

En el proceso de la HUM-08 (Gestión de Beneficios) descrita en la Tabla 10, se creó un Seeder con los beneficios que ya ofrece la organización, lo cual trata sobre los pagos anuales, inscripción de un nuevo socio tomando en cuenta ciertas características como edad, tiempo de casados, estado civil, etc. Este apartado se lo desarrolló no solo mediante una inserción de datos por medio del Seeder, sino como CRUD para que se puedan ingresar nuevos beneficios en caso de que la organización lo necesite. Como se muestra en la Figura 21.

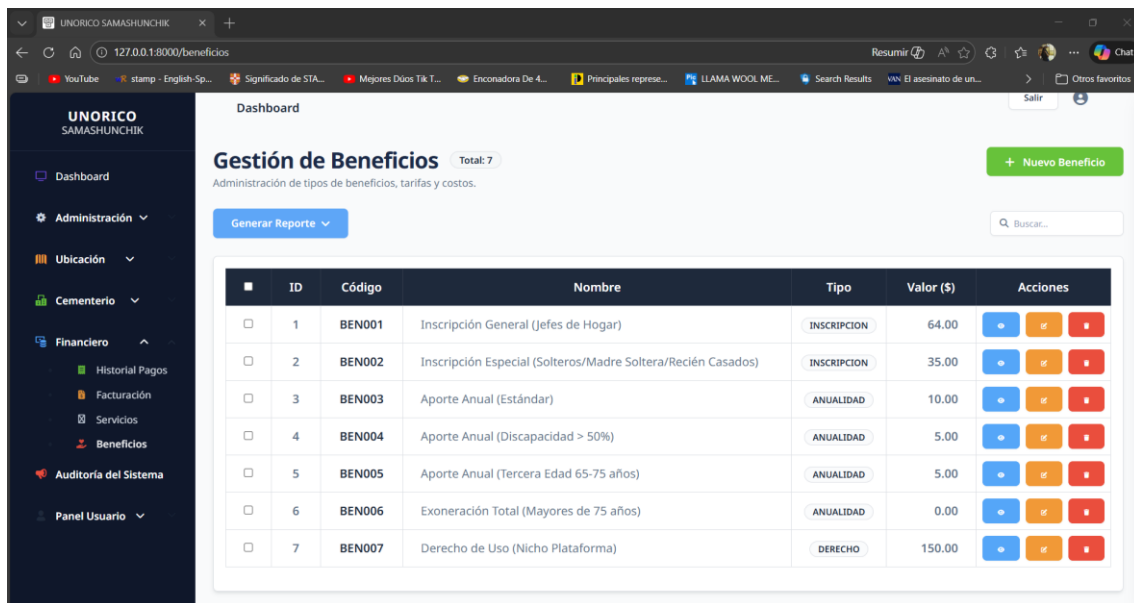


Figura 21 Gestión de Beneficios

Continuamos con el desarrollo de la HUM-09 (Gestión de Socios) detallada en la Tabla 11, el backend se lo realizó en el archivo SocioController, aquí es donde se pone toda la lógica del CRUD de cómo deben de funcionar cada método para la gestión de los socios. El frontend se muestra con su tabla index donde se pueden ver los registros de los socios con sus respectivas columnas y acciones. Como se indica en la Figura 22.

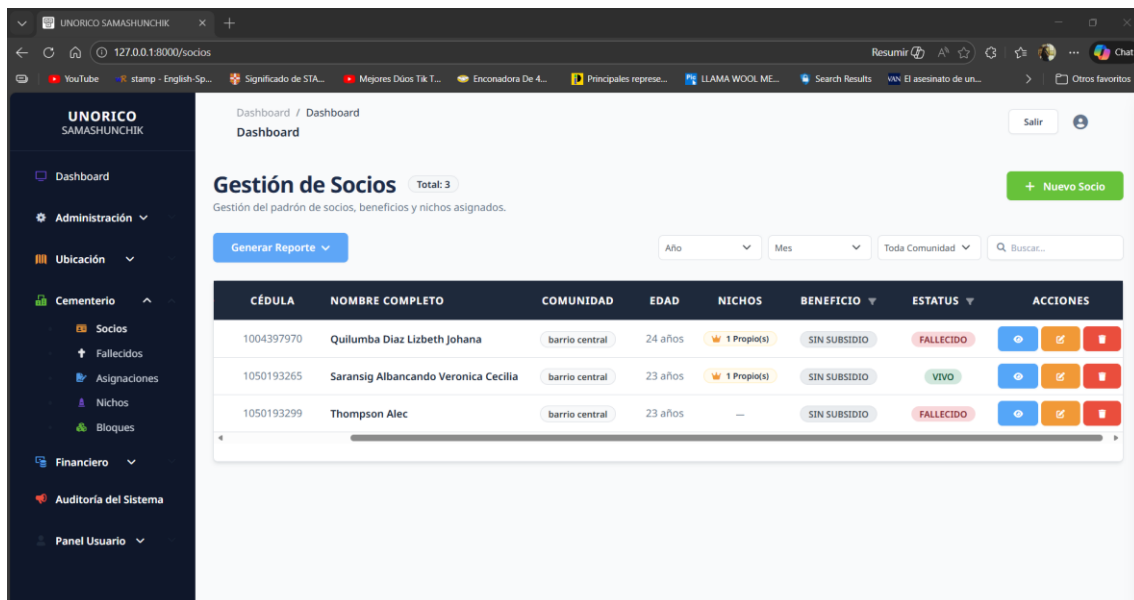
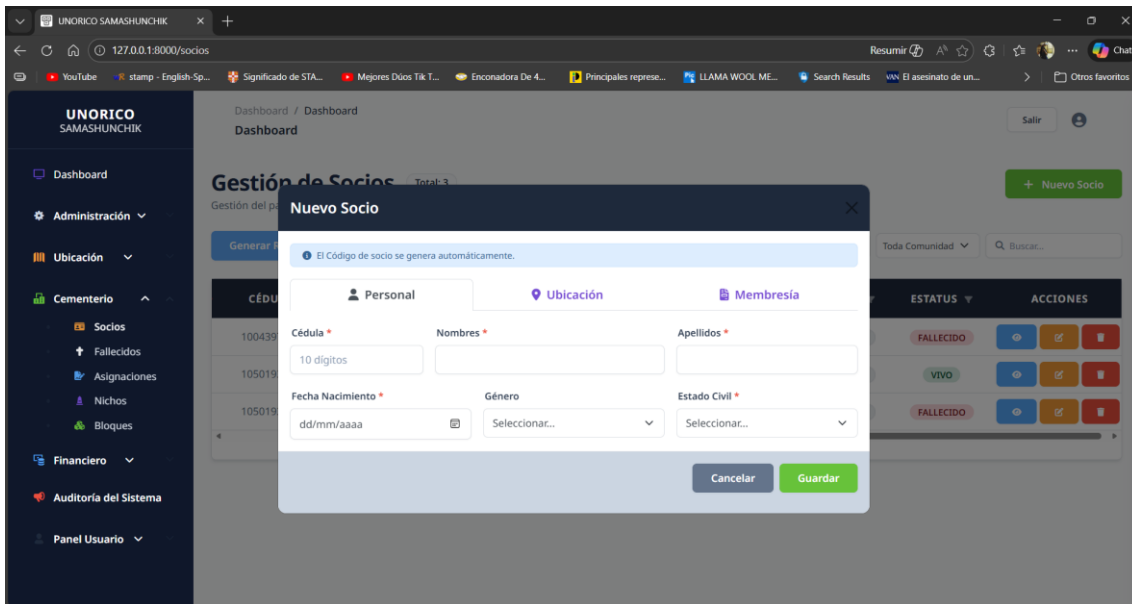


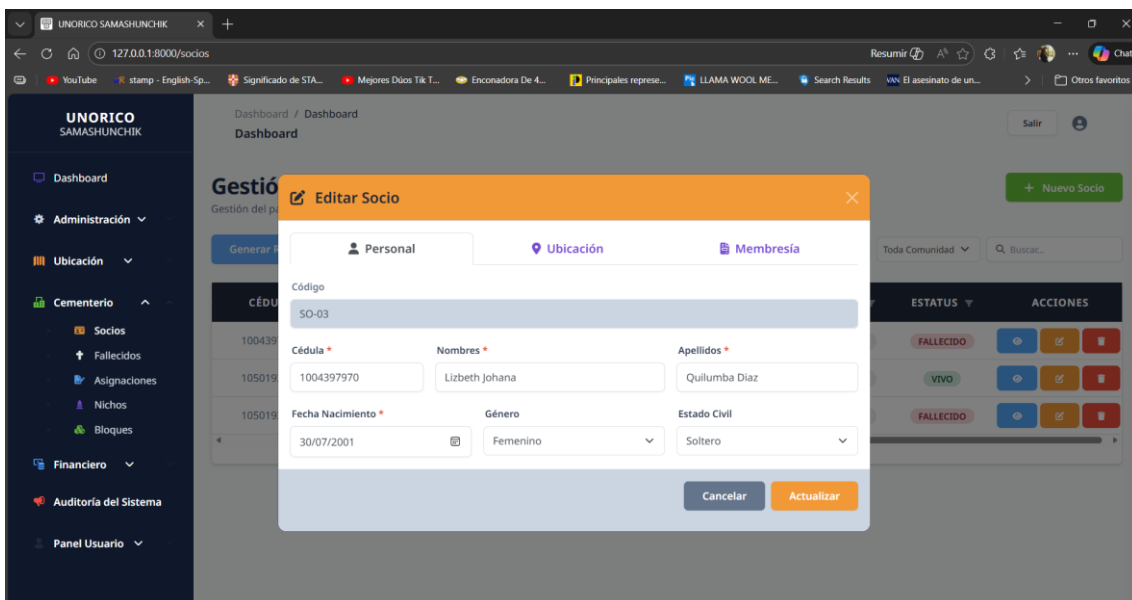
Figura 22 Gestión de Socios

Para crear un nuevo socio se hizo un modal con un formulario para ingresar los datos para registrar al socio. Esto se indica en la Figura 23.



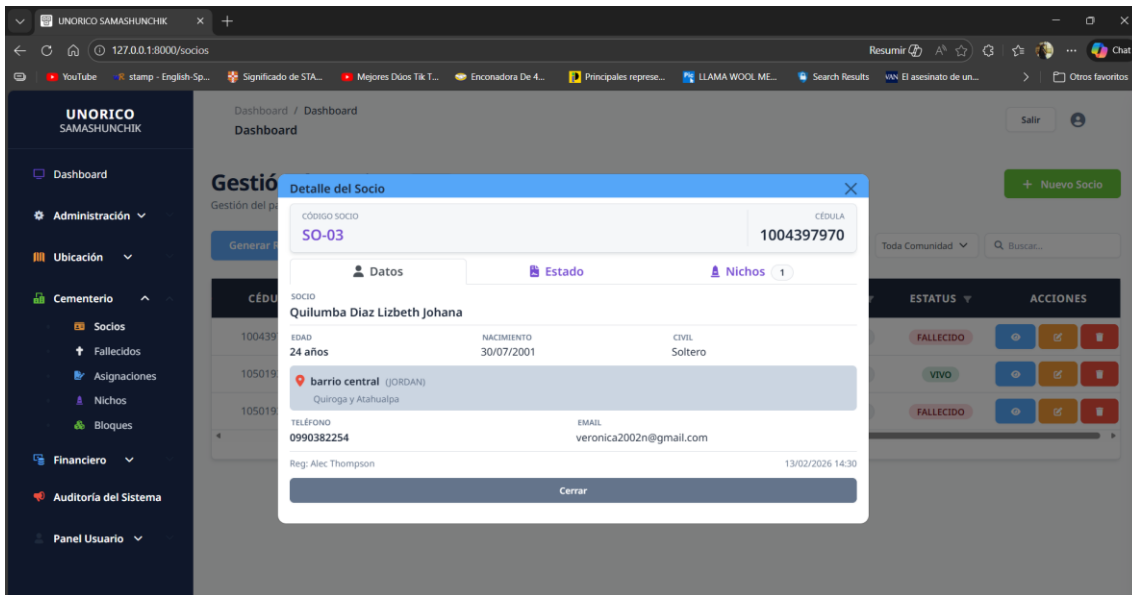
*Figura 23 Nuevo Socio*

Para editar un socio ya existente vamos a la tabla y damos clic en el icono del lápiz y se abrirá un modal con su respectivo formulario y podremos editar cualquier registro que hayamos seleccionado, cambiamos algún dato y damos clic en actualizar y nos saldrá en la vista Index un mensaje en verde de socio actualizado correctamente. Como se ve en la Figura 24.



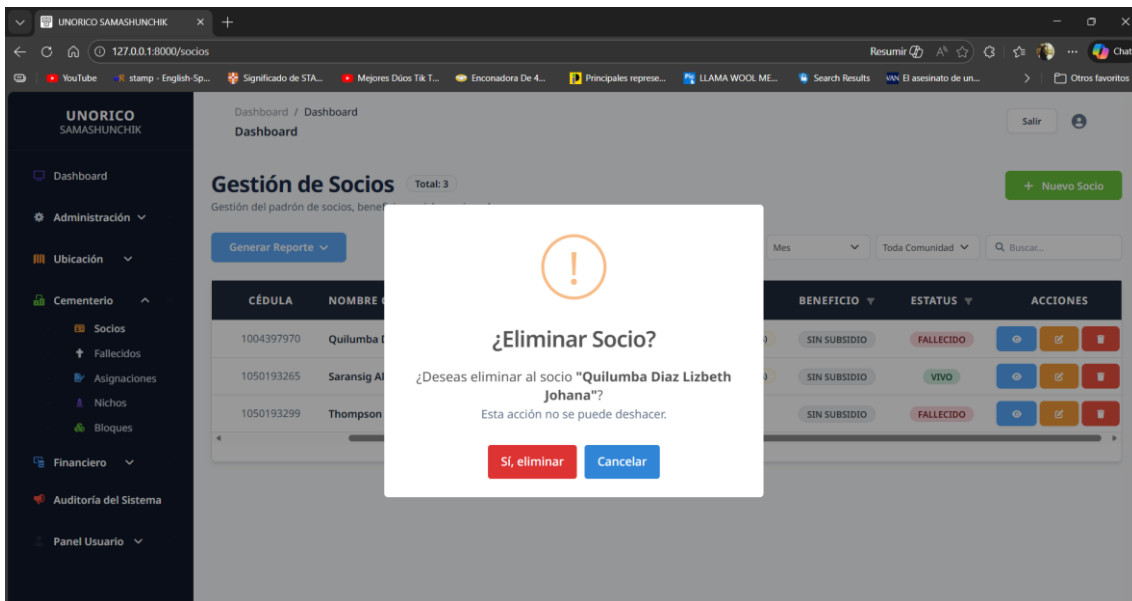
*Figura 24 Editar / Actualizar Socio*

En caso de que solo se quiera ver la información completa del socio, solo damos clic en el icono del ojo y se nos abrirá un modal, pero de tipo lectura, lo cual solo permite leer los datos que ya están ingresados. Esto se ilustra en la figura 25.



*Figura 25 Detalle del Socio*

Y por última acción que se puede realizar es eliminar a un socio, para hacer esto se da clic en el icono del basurero y nos saldrá un mensaje de confirmación para estar seguros de que deseamos eliminar a ese socio. Como se aprecia en la Figura 26.



*Figura 26 Eliminar Socio*

Seguimos con la HUM-10 (Buscar y Filtrar Socios), correspondiente a la Tabla 12, se incorporó Query Scopes en el modelo de la base de datos para facilitar la búsqueda en tiempo real mediante cédula, nombres y apellidos. Lo eficaz de esta búsqueda se comprobó en la vista principal Gestión de Socios donde se integró un buscador por medio de peticiones asíncronas AJAX, el cual filtra e indica instantáneamente los registros que

contiene la base de datos sin requerir recargar la página. Igualmente, para filtrar a ciertos socios, lo que se integró es un filtro como año, mes y comunidad. Como se indica en la Figura 27.

Dashboard / Dashboard

**Gestión de Socios** Total: 2

Salir

+ Nuevo Socio

Generar Reporte

Año Mes Toda Comunidad 1050

#	CÓDIGO	CÉDULA	NOMBRE COMPLETO	COMUNIDAD	EDAD	NICHOS	BENEFICIO	ESTATUS
1	SO-01	1050193265	Saransig Albancando Veronica Cecilia	barrio central	23 años	1 Propio(s)	SIN SUBSIDIO	VIVO
2	SO-02	1050193299	Thompson Alec	barrio central	23 años	—	SIN SUBSIDIO	FALLECIDO

*Figura 27 Buscar Socio en tiempo real*

Con respecto a la HUM-11 (Control de Exonerados), expuesta en la Tabla 13, lo principal en desarrollar fue la lógica del backend para lograr automatizar el cumplimiento del Art. 24 del Reglamento Interno de la Organización UNORICO para socios mayores de 75 años [61]. En el backend se realizó una validación que no permite asignar el beneficio de “Exonerado” si el socio no ha cumplido los 75 años o más, en caso de cumplirse se debe de poner la fecha en la cual fue exonerado. También se programaron notificaciones que nos muestran sobre los socios que ya han cumplido la edad y que no han sido procesados, lo cual nos permite actualizarlo. Esto se ilustra en la Figura 28.

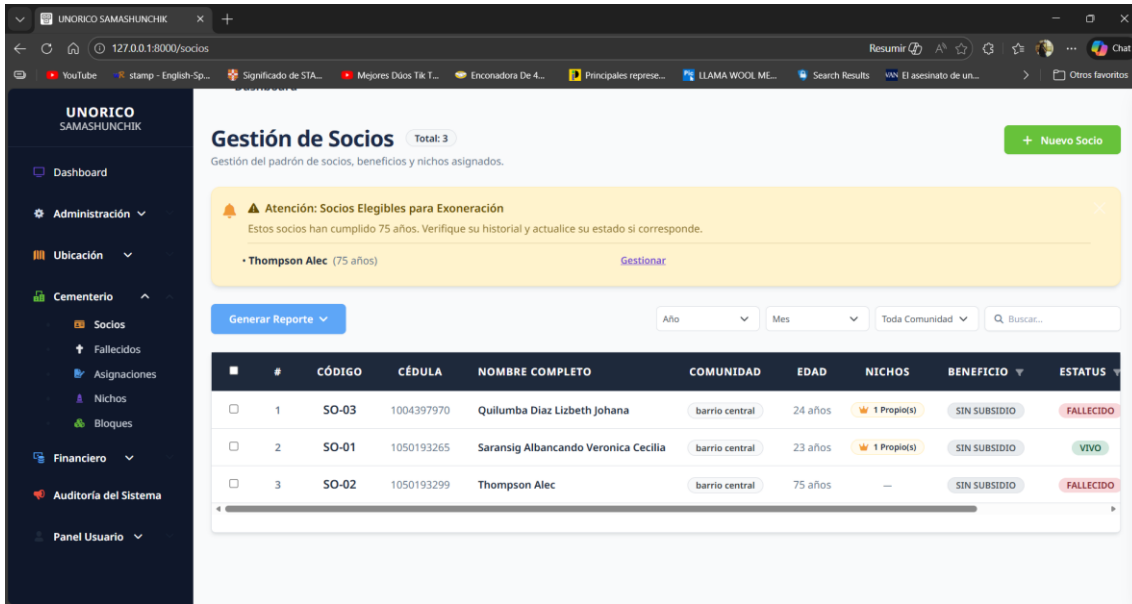


Figura 28 Control de Socios Exonerados

Finalmente, para la HUM-12 (Reporte de Socios), detallada en la Tabla 14, lo que se hizo fue exportar los datos en formatos PDF y Excel generando sus respectivos reportes. También se implementó la lógica para que capture los filtros que hay en la pantalla principal (año, mes y comunidad) y ciertos registros seleccionados a través de *checkboxes* para generar un documento enfocado en lo que se requiere. Se validó que la descarga del archivo incluya la fecha y hora en el nombre. Como se ve en la Figura 29.

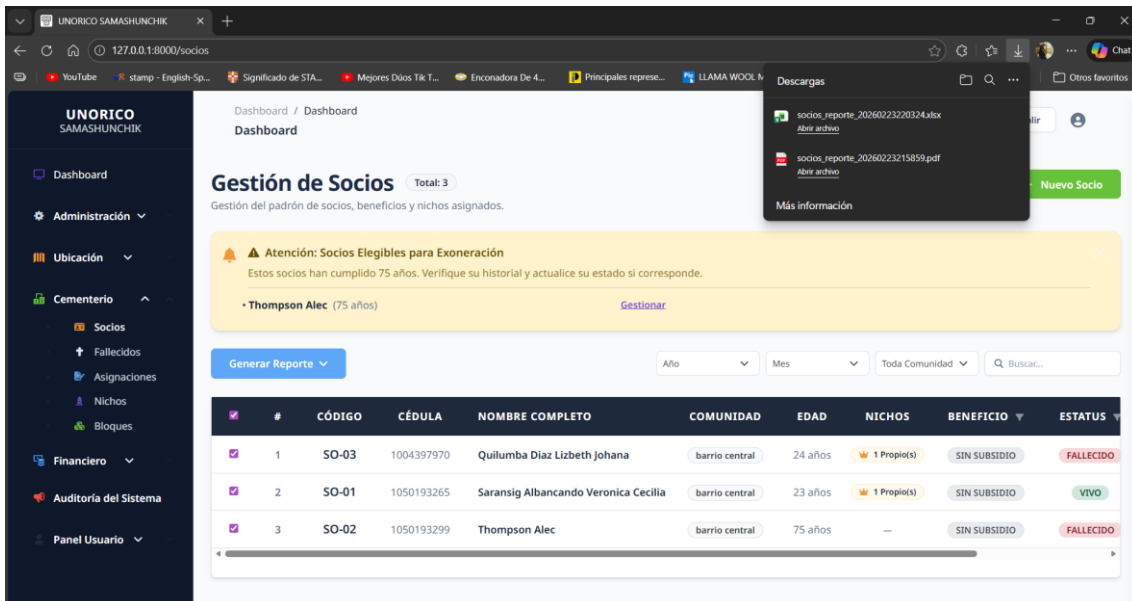


Figura 29 Descargar Reporte Socios PDF y Excel

A continuación, verificamos la descarga del documento tanto en PDF como en Excel y su respectivo contenido. Esto se ilustra en las Figuras 30 y 31.

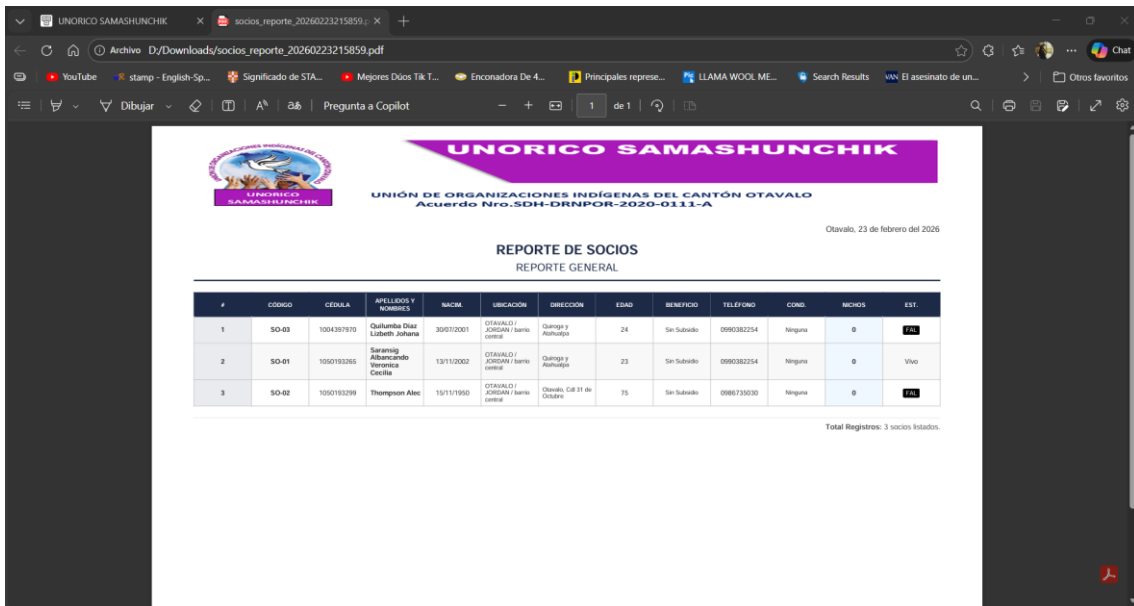


Figura 30 Reporte Socios PDF

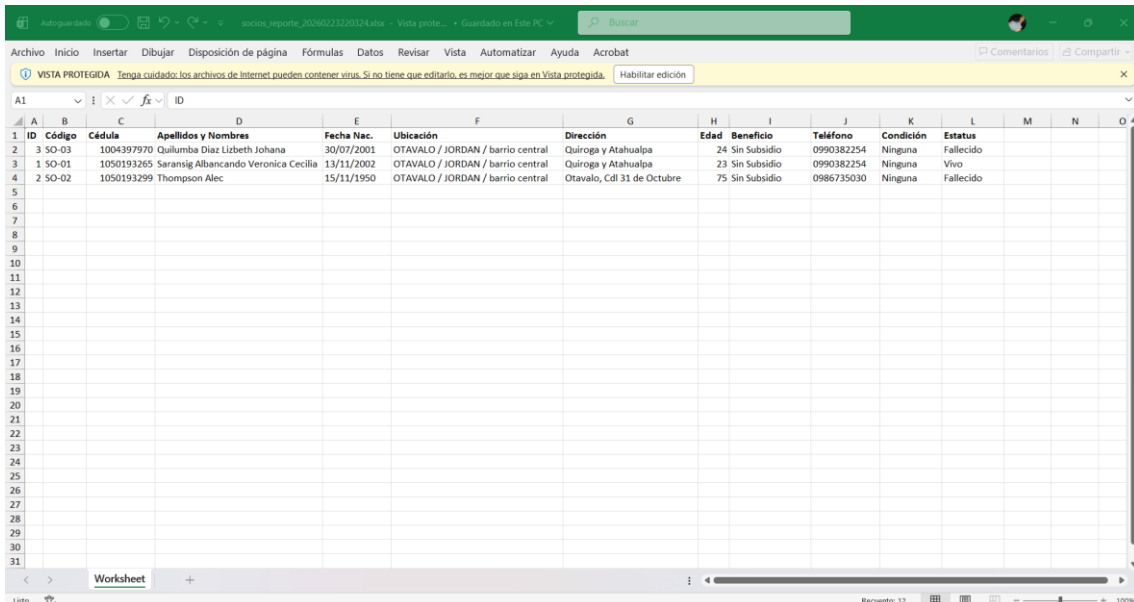


Figura 31 Reporte Socios Excel

- Revisión – Sprint 2

A continuación, se describe el cumplimiento de los criterios de aceptación de las Historias de Usuario que fueron desarrolladas en este segundo Sprint. Se pueden observar estos criterios en la Tabla 38.

Tabla 38 Revisión - Criterios de Aceptación Sprint 2

Historias de Usuario	Criterio de Aceptación	Cumple
----------------------	------------------------	--------

<b>Gestión de Servicios</b>	CA1: La aplicación debe permitir crear un servicio con los campos: nombre (obligatorio) , descripción (opcional), precio (numérico).	SI
	CA2: El código único de servicio se genera automáticamente cuando se crea un nuevo servicio,	SI
	CA3: La aplicación debe permitir editar un servicio modificando todos los campos que se encuentran.	SI
	CA4: La aplicación debe de permitir eliminar un servicio. Cuando se elimina mostrar un mensaje “Servicio Eliminado”.	SI
	CA5: El buscar tiene que filtrar por nombre o código en tiempo real.	SI
	CA6: Se debe de seleccionar todos los servicios o algunos para generar un reporte en formato PDF y Excel.	SI
	CA7: En caso de no seleccionar ningún servicio y tratar de generar el reporte se indicará un mensaje “Debe de seleccionar al menos un servicio”.	SI
<b>Gestión de Beneficios</b>	CA1: La aplicación debe permitir crear un beneficio con los campos: nombre (obligatorio), descripción (opcional), tipo (obligatorio) y valor ( obligatorio, numérico).	SI
	CA2: El código único de beneficios se genera automáticamente cuando se crea un nuevo beneficio.	SI
	CA3: La aplicación debe permitir editar un beneficio modificando todos los campos que se encuentran.	SI
	CA4: La aplicación debe de permitir eliminar un beneficio. Cuando se elimina mostrar un mensaje “Beneficio Eliminado”.	SI
	CA5: El buscar tiene que filtrar por nombre o código en tiempo real.	SI
	CA6: Se debe de seleccionar todos los beneficios o algunos para generar un reporte en formato PDF y Excel.	SI
	CA7: En caso de no seleccionar ningún beneficio y tratar de generar el reporte se indicará un mensaje “Debe de seleccionar al menos un beneficio”.	SI
<b>Gestión de Socios</b>	CA1: La aplicación tiene que permitir crear un socio con los siguientes campos: Cédula, Nombres, Apellidos, Fecha de Nacimiento, Teléfono, Dirección, Email, Comunidad, Estado Civil, Género, Fecha de Inscripción, Tipo de Beneficio (Sin Subsidio / Con Subsidio / Exonerado), Condición (Ninguna / Discapacidad / Enfermedad Terminal) y Estatus (Vivo / Fallecido).	SI

	CA2: Se debe de tomar en cuenta que estos campos son obligatorios: Cédula, Nombres, Apellidos, Fecha de Nacimiento, Comunidad, Estado Civil, Fecha de Inscripción, Tipo de Beneficio, Condición y Estatus.	SI
	CA3: La cédula debe ser única. En caso de repetirse mostrar el mensaje “Este número de cédula ya existe. El socio ya esa registrado”.	SI
	CA4: Si el tipo de beneficio es “Exonerado” debe de tomarse en cuenta que la edad del socio sea mayor o igual a 75 años. Si no cumple con esto mostrar el mensaje “No se puede crear como exonerado”.	SI
	CA5: Si el tipo de beneficio es “Exonerado” y cumple con el requisito de mayor o igual a 75 años se debe de poner obligatoriamente la fecha de exoneración.	SI
	CA6: El código del socio se genera automáticamente cuando se registra.	SI
	CA7: Cuando se crea exitosamente indicar el mensaje “Socio creado correctamente xxxx”.	SI
	CA8: La aplicación debe permitir editar todos los campos del socio	SI
	CA9: La validación del beneficio “Exonerar” por edad mínima 75 años también se puede editar.	SI
	CA10: Cuando se actualiza exitosamente los datos de un socio debe de mostrar un mensaje “Socio actualizado correctamente”.	SI
	CA11: Cuando se ve el detalle de un socio se debe de indicar todos sus datos personales y los nichos que este tiene con sus respectivas ubicaciones (bloque / nicho).	SI
	CA12: La aplicación debe de solicitar una confirmación antes de eliminar un socio.	SI
	CA13: Al eliminar exitosamente se muestra un mensaje “Socio eliminado correctamente”.	SI
<b>Buscar y Filtrar Socios</b>	CA1: La aplicación debe tener una barra de búsqueda que filtre en tiempo real por: Cédula, Apellidos, Nombres o Código del socio.	SI
	CA2: Los resultados de la búsqueda deben de actualizarse dinámicamente, cuando el usuario se encuentre escribiendo con un icono de carga.	SI
	CA3: La aplicación tiene que permitir filtrar a los socios por comunidades por medio de un select desplegable.	SI
	CA4: La aplicación debe permitir filtrar por estatus seleccionado (vivo o fallecido).	SI

	CA5: La aplicación tiene que permitir filtrar por tipo de beneficio escogiendo varias opciones (Subsidio, Sin subsidio, Exonerado).	SI
	CA6: La aplicación debe filtrar por mes y año de inscripción o registro.	SI
	CA7: La tabla donde se encuentra el listado debe de tener los campos: código, cédula, nombres y apellidos, comunidad, tipo de beneficio, estatus, y el conteo de nichos (total, propios y compartidos).	SI
	CA8: Los registros deben de estar con una paginación de 10 por página, conservando los filtros cuando se cambie de página.	SI
<b>Control de Exonerados</b>	CA1: La aplicación nos indica una alerta en el apartado de gestión de socios cuando exista personas para exoneración (socios con 75 años o más que aún no tengan el beneficio de exonerado).	SI
	CA2: Esta alerta debe de mostrar la cantidad de socios que han sido identificados para exonerar.	SI
	CA3: Al crear o editar un socio con el beneficio “Exonerado” se debe de validar que cumpla con el requisito “mayor o igual a 75 años” caso contrario muestra un mensaje “No se puede crear como exonerado”.	SI
	CA4: Cuando se el beneficio se Exonerado y cumpla con el requisito la fecha de exoneración es obligatorio.	SI
	CA5: La aplicación debe de calcular la edad del socio automáticamente con la fecha de nacimiento.	SI
	CA6: Se tiene que poder filtrar los socios por el estado de beneficio “Exonerado” para visualizar cuales son los socios exonerados,	SI
<b>Reporte de Socios</b>	CA1: La aplicación tiene que permitir seleccionar todos lo algunos socios mediante checkboxes en la tabla.	SI
	CA2: Se debe de poder generar un reporte en formato PDF o Excel con sus respectivos campos.	SI
	CA3: El reporte en formato PDF debe de ser en formato A4 horizontal para visualizar correctamente todos los datos.	SI
	CA4: Los filtros que se tiene en la tabla como estatus y beneficio al momento de generar el reporte deben de aplicarse.	SI
	CA5: En caso de filtrar por mes / año en el reporte se debe de poner un subtítulo indicando el tipo de filtro que se aplicó.	SI
	CA6: Sino se selecciona ningún socio se muestra un mensaje “Debe seleccionar al menos un socio para generar el reporte”.	SI

- Retrospectiva – Sprint 2

En la retrospectiva del Sprint 2, se enfatizó el fortalecimiento del entorno de servicios, beneficios y la gestión de los socios. Durante este proceso, se desarrollaron los apartados de Gestión de Servicios y Beneficios, los cuales permiten administrar de mejor manera las prestaciones que brindan. Un hito técnico importante fue el desarrollo de la Gestión de Socios, incorporando un motor de búsqueda en tiempo real; esto mejora la localización de los socios registrados. Igualmente se mejoró el Control de Exonerados, haciendo posible aplicar reglas de negocio para los socios mayores o iguales a 75 años. La finalización de este sprint es generar reportes de socios en formatos PDF y Excel para que la directiva tenga un enfoque claro acerca del padrón de asociados, simplificando la toma de decisiones respaldada en datos precisos.

### 3.2.4 Sprint 3

- Planificación Sprint 3

En el tercer sprint se enfocó en el desarrollo para la gestión del espacio físico y la gestión de los fallecidos y asignaciones. En esta fase se construyó la lógica para gestionar y controlar los bloques y nichos para lograr identificar dónde están sepultados los fallecidos y saber la disponibilidad del espacio. De igual manera se realizaron los procesos de Inhumación (Entierro) que en la aplicación están como asignación y también la Exhumación mediante esto se vincula a cada fallecido a un responsable (socio) y se designa una ubicación (bloque/nicho). En este sprint contendrá las historias de usuario desde la HUM-13 a la HUM-20

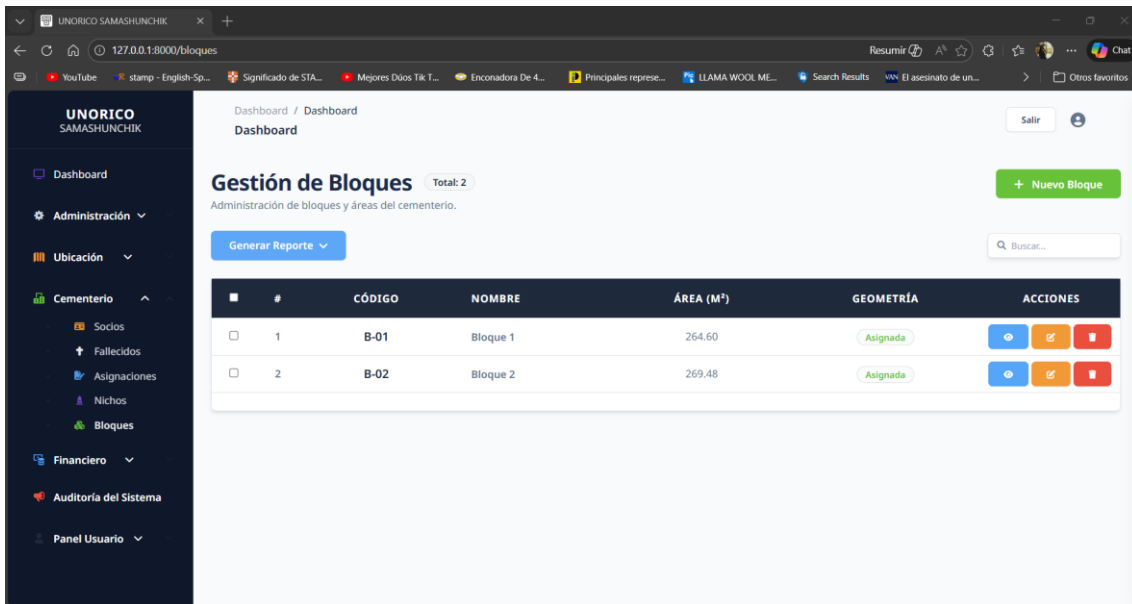
*Tabla 39. Sprint Backlog – Sprint 3*

<b>Historia / Código</b>	<b>Nombre</b>	<b>Tarea</b>
<b>HUM-13</b>	Gestión de Bloques	<p><b>Base de Datos:</b> Crear tabla bloques con campos: nombre, descripción, bloques_geom_id.</p> <p><b>Backend:</b> Controlador BloqueController con métodos store/update/destroy.</p>

		<b>Frontend:</b> Diseñar vista bloques.index con tabla de lista y Modales para Crear/Editar bloque.
<b>HUM-14</b>	Gestión de Nichos	<p><b>Base de Datos:</b> Crear tabla nichos vinculada a bloques, nichos_geom y con estado por defecto 'DISPONIBLE'.</p> <p><b>Backend:</b> Lógica para que pueda seleccionar un nicho con identificador GIS y con su respectivo CRUD.</p> <p><b>Frontend:</b> Formulario de registro de Nicho y Vista de administración (Tabla) con filtros por estado.</p>
<b>HUM-15</b>	Ver Mapa Cementerio	<p><b>Frontend / Vista:</b> Integración de un hipervínculo en la vista show que redirige al usuario a una nueva ventana para visualizar la ubicación.</p> <p><b>Backend:</b> No aplica. Se utiliza una redirección estática hacia un servicio de mapas externo en internet, sin necesidad de procesar datos espaciales en el servidor.</p>
<b>HUM-16</b>	Gestión Fallecidos	<p><b>Base de Datos:</b> Tabla fallecidos (independiente de socios). Campos: nombres, fecha_nacimiento, fecha_deceso, etc.</p> <p><b>Backend:</b> FallecidoController para gestionar la información civil del difunto.</p> <p><b>Frontend:</b> Formulario de registro de defunción mediante modales.</p>
<b>HUM-17</b>	Buscar Fallecidos	<p><b>Backend:</b> Implementar filtros de búsqueda por nombre, cédula o fecha en el controlador.</p> <p><b>Frontend:</b> Barra de búsqueda en la vista principal de fallecidos con filtrado AJAX.</p>

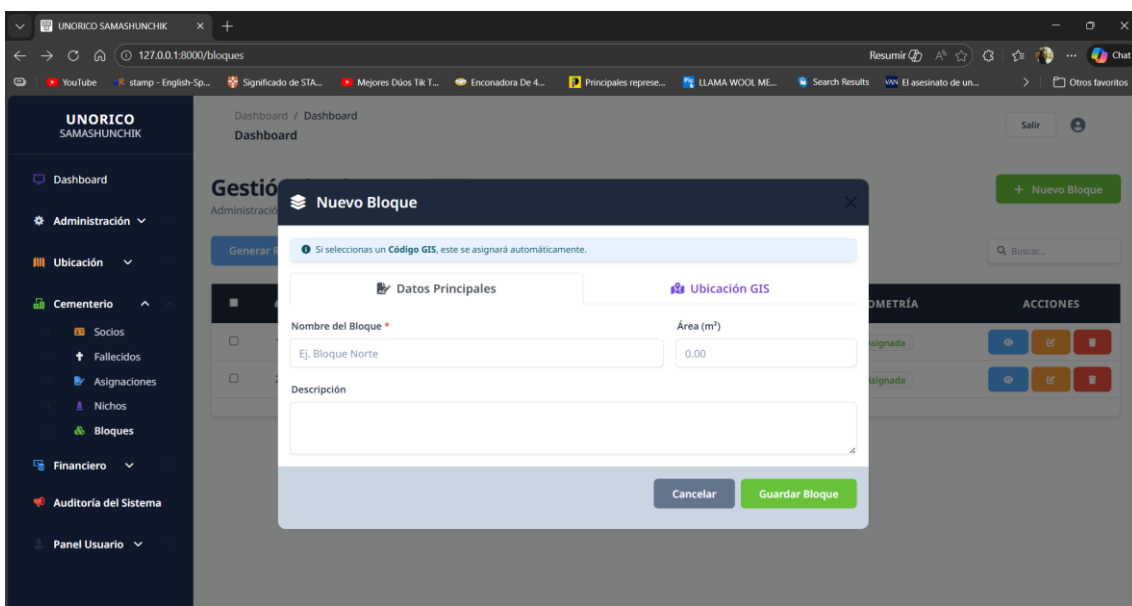
<b>HUM-18</b>	Inhumación (Asignar)	<b>Base de Datos:</b> Crear tabla pivote fallecido_nicho con sus respectivos campos (fallecido_id, nicho_id, socio_id, fecha_inhumacion), etc.
		<b>Backend:</b> Transacción DB: Crear registro inhumación + Actualizar estado del Nicho a 'OCUPADO' y actualizar la ocupación.
		<b>Frontend:</b> Vista modal para el registro de una nueva asignación ingresando fallecido, nicho y responsable (socio).
<b>HUM-19</b>	Exhumación	<b>Backend:</b> Lógica para finalizar asignación, guardar fecha salida y cambiar estado Nicho a 'DISPONIBLE' y disminuir automáticamente la ocupación.
		<b>Frontend:</b> Botón "Registrar Exhumación" en ver más detalles del fallecido con confirmación de seguridad.
<b>HUM-20</b>	Reporte de Asignaciones	<b>Backend:</b> Generación de reporte PDF con historial de inhumaciones por rango de fechas.
		<b>Frontend:</b> Interfaz para seleccionar rango de fechas (Desde - Hasta) y botón "Generar Reporte".

Para la HUM-13 (Gestión de Bloques), detalla en la Tabla 15, su desarrollo se centró en integrar la gestión administrativa con información geográfica mediante PostGIS. En el backend se realizó un CRUD que permite conectar un registro de datos de un polígono incorporado desde QGIS, la aplicación detecta si hay una conexión para obtener el código del mapa o generar uno interno, esto calcula automáticamente el área en metros cuadrados. En cambio, en el frontend la administración se lleva a cabo por modales AJAX que están organizados por pestañas y selectores dinámicos. Esto se puede observar en la Figura 32.

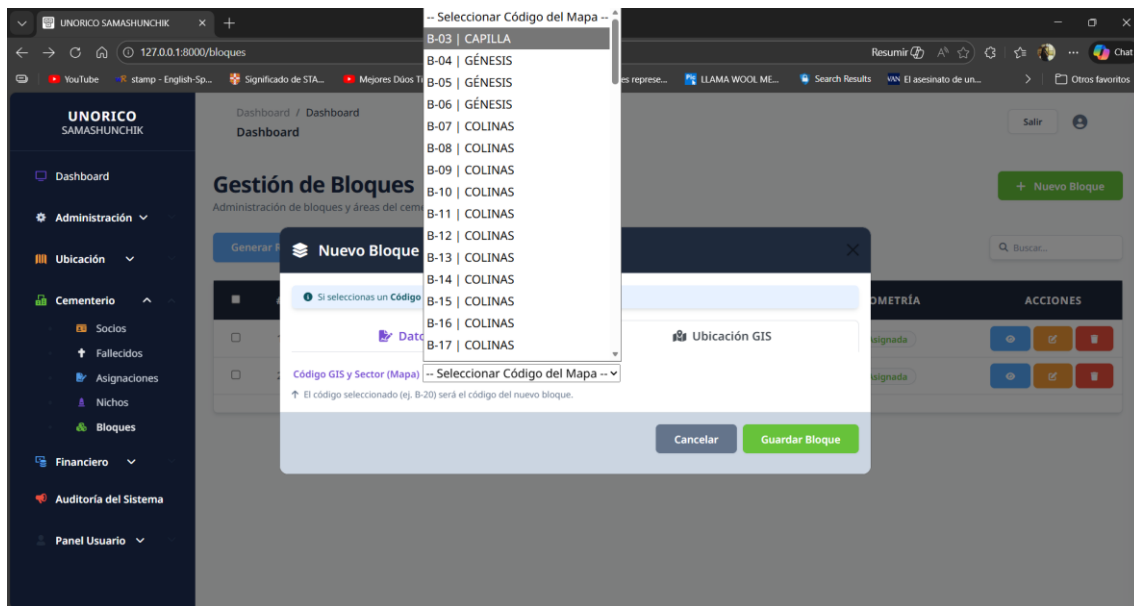


*Figura 32 Gestión Bloques*

Para crear un nuevo bloque se da clic en el botón “Nuevo Bloque” donde nos indicará el modal y las pestañas mencionadas anteriormente para empezar a llenar datos y seleccionar el bloque que ya fue dibujado en QGIS. Estas vistas se presentan en las Figuras 33-34.



*Figura 33 Crear Bloque pestaña Datos*



*Figura 34 Crear Bloque pestaña ubicación GIS*

También en este apartado se puede sacar un reporte de los bloques que existen o seleccionar ciertos bloques de igual manera se controló que una vez seleccionado un bloque de GIS para crear un nuevo bloque en la aplicación, este no vuelve a repetirse en la lista. Esto se hizo para no generar errores en las ubicaciones.

Continuando con la HUM-14 (Gestión de Nichos), correspondiente a la Tabla 16, se orientó a la administración individual los nichos que están dentro de los bloques ya creados. Se implementó un CRUD completo el cual permite relacionar a cada nicho con un bloque y con un socio responsable, también nos deja seleccionar un código GIS como en la gestión de bloques. Algo nuevo que se realizó fue el desarrollo de generación de Códigos QR de modo texto, lo cual permitirá a los socios solicitar el QR a la administración para colocarlo en el nicho y se pueda visualizar la información necesaria como: nombres de los fallecidos, bloque y nicho al que pertenecen y el socio que se encuentra a cargo. Todo esto se ilustra en las Figuras 35-38.

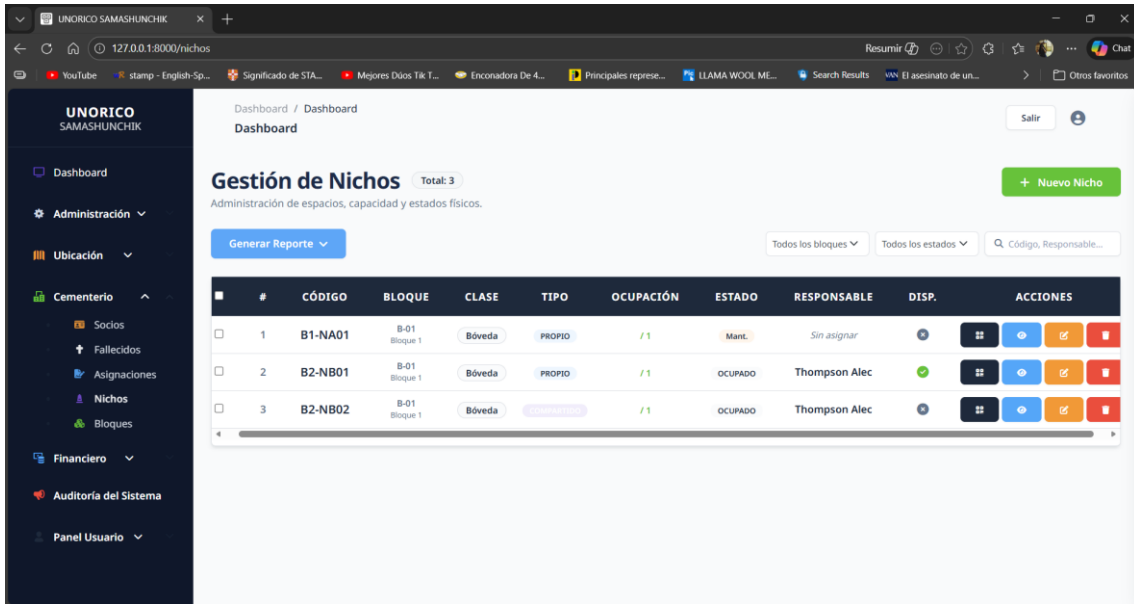


Figura 35 Gestión Nichos

Cuando se crea un nuevo nicho se debe de llenar dos pestañas: ubicación en donde se de seleccionar el nicho, bloque y socio y la pestaña datos técnicos para ver las especificaciones del nicho en caso de que comparta este con familia u x persona y si es que es propio. Como se indica en las Figuras 36-38

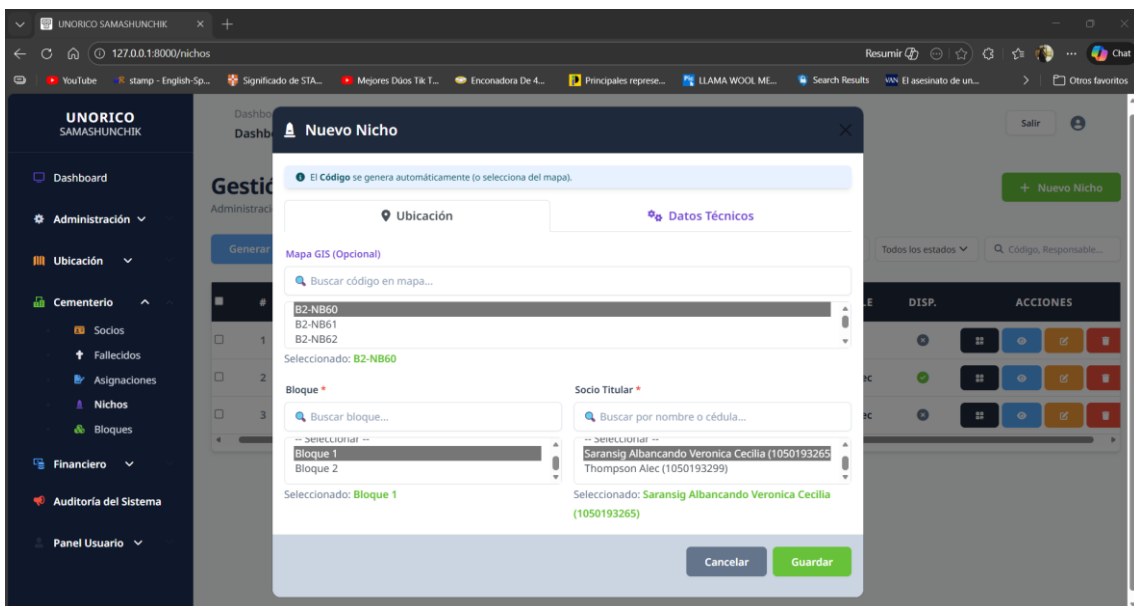
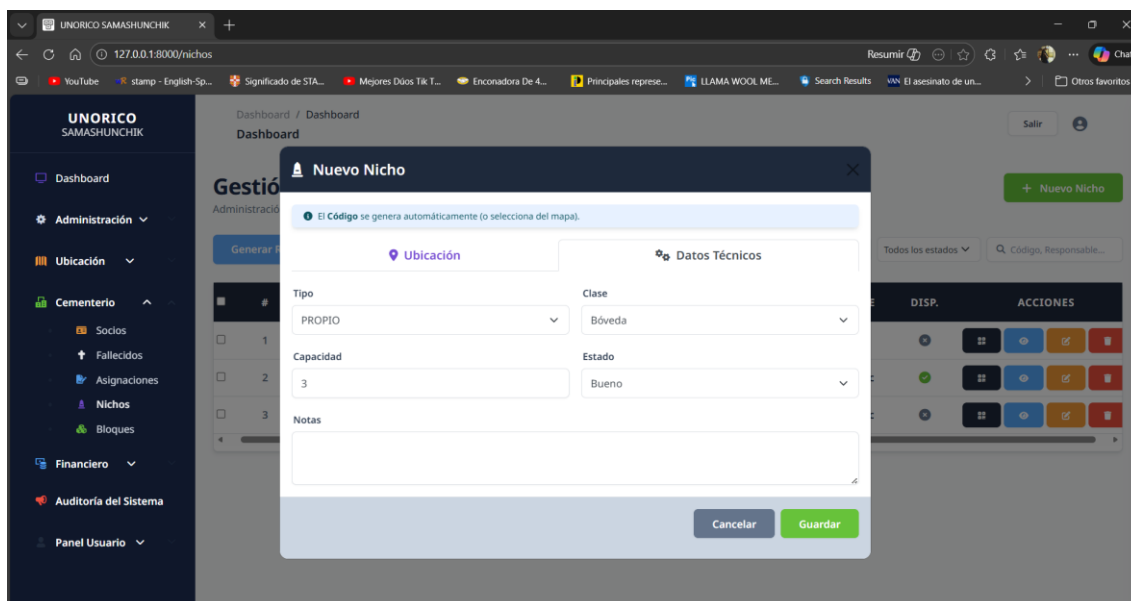
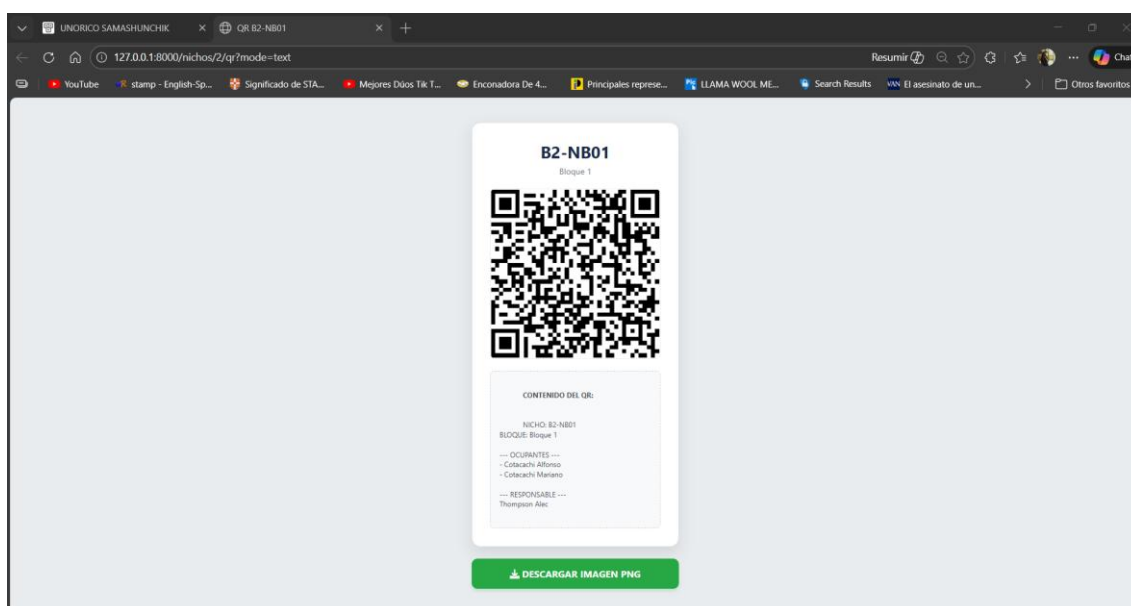


Figura 36 Crear Nicho pestaña Ubicación



*Figura 37 Crear Nicho pestaña Datos Técnicos*

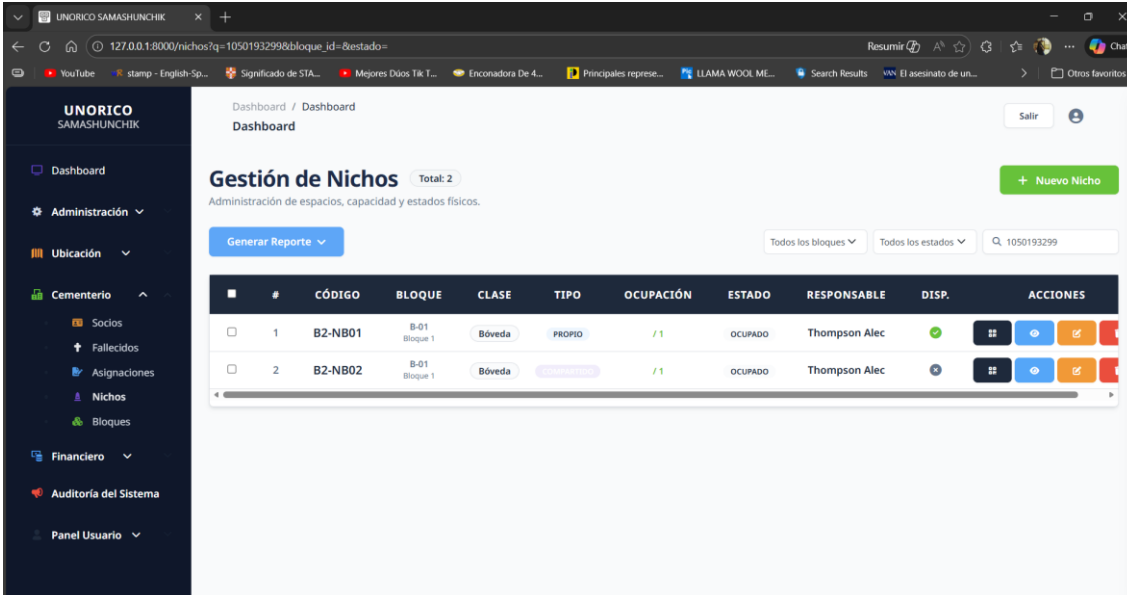
Con respecto al Código QR cuando se asigne un fallecido con sus respectivos datos se visualizará la información en el QR, para generar este debemos dar clic en el botón color azul con icono de cuadros, este abrirá una nueva pestaña indicando el código, qué información contiene y descargar imagen PNG. Esto se visualiza en la Figura 38.



*Figura 38 Código QR*

De igual forma este apartado contiene la opción de generar reportes en archivos PDF y Excel mediante sus respectivos filtros como: por bloques o según el estado físico del nicho (mantenimiento, bueno, malas condiciones y abandonado) o por medio de selección de algunos nichos también se realiza la búsqueda de un cierto nicho en tiempo real mediante la cedula del socio asignado o sus nombres para identificar cuantos nichos

tiene el socio de que tipo son, que clase y como se encuentran estos. Esto se refleja en la Figura 39.



Dashboard / Dashboard

Dashboard

Gestión de Nichos Total: 2

Administración de espacios, capacidad y estados físicos.

Generar Reporte

Todos los bloques Todos los estados 1050193299

#	CÓDIGO	BLOQUE	CLASE	TIPO	OCUPACIÓN	ESTADO	RESPONSABLE	DISP.	ACCIONES
1	B2-NB01	B-01 Bloque 1	Bóveda	PROPIO	/ 1	OCUPADO	Thompson Alec		
2	B2-NB02	B-01 Bloque 1	Bóveda	OCUPADO	/ 1	OCUPADO	Thompson Alec		

Figura 39 Búsqueda de nichos

Siguiendo con la HUM-15 (Ver Mapa Cementerio) mencionada en la Tabla 17, la implementación se concentró en cómo sería compatible la aplicación web con un sistema de información geográfica externo. Lo que se realizó fue conectar la aplicación desarrollada en Laravel con el mapa que se encontraba desplegado en Render.com. Funcionalmente, se incorporó un botón “Ubicar Mapa” dentro del modal Show (detalles de Asignaciones) cuando se lo presiona, la aplicación valida si el nicho tiene un código GIS y abre otra pestaña con la URL donde está alojado el mapa del catastro. Como se ilustra en la Figura 40-41.

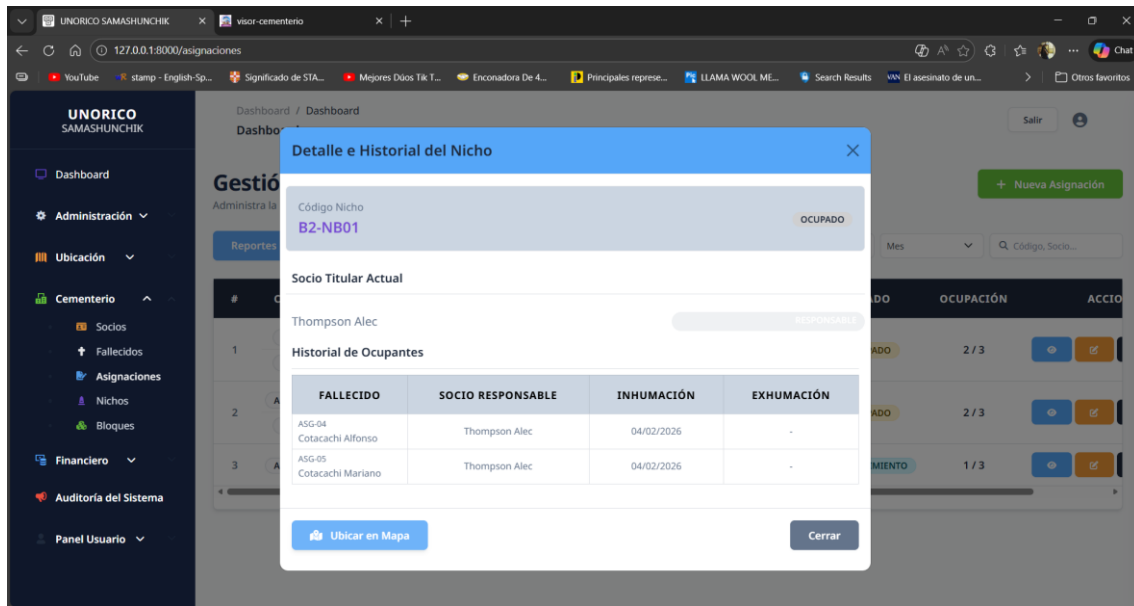


Figura 40 Botón Ubicar Mapa

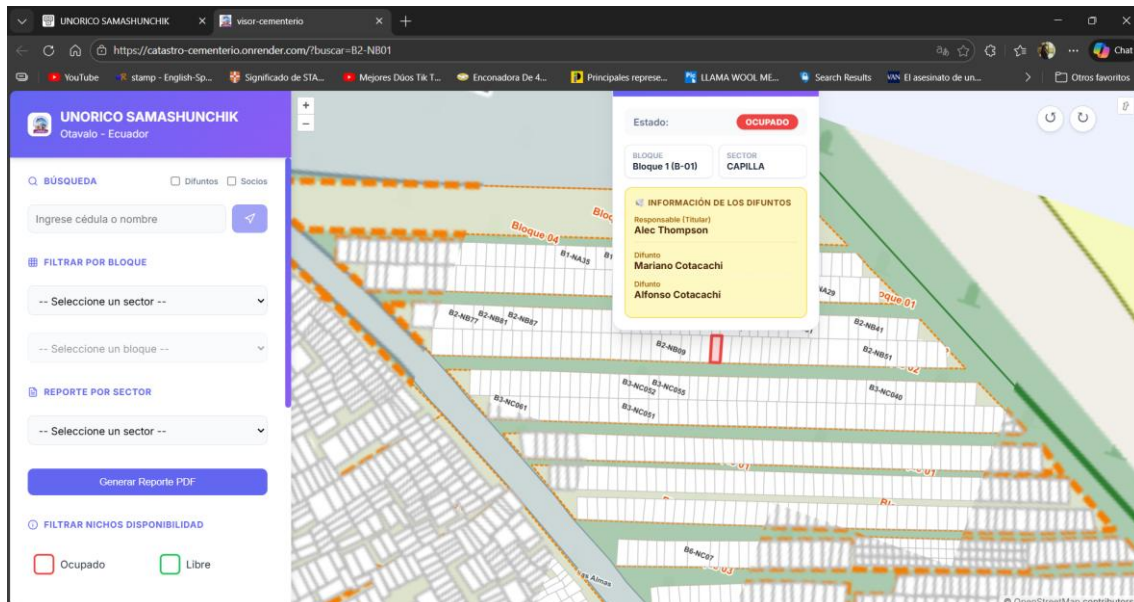


Figura 41 Mapa del Catastro con ubicación de un nicho con asignación

Para el desarrollo de la HUM-16 (Gestión de Fallecidos), referenciada en la Tabla 18, se realizó el desarrollo del apartado de gestión de fallecidos con la capacidad de sincronización automática. En el backend se programó una lógica de negocio el cual cuando se registre un nuevo fallecido y se ingrese un número de cédula esta consulta en tiempo real a la base de datos de Socios, si existe una coincidencia en la parte de fallecidos se autocompleta los campos de las pestañas y solo se debe de ingresar el campo de fecha de fallecido y causa se guarda el registro y en la gestión de Socios el campo “Estatus” automáticamente cambia a “Fallecido”, lo que garantiza la integridad en ambos apartados.

Igualmente cuenta con la sección para filtrar y buscar a los fallecidos, así como realizar reportes en PDF y Excel. Esto se ilustra en las Figura 42.

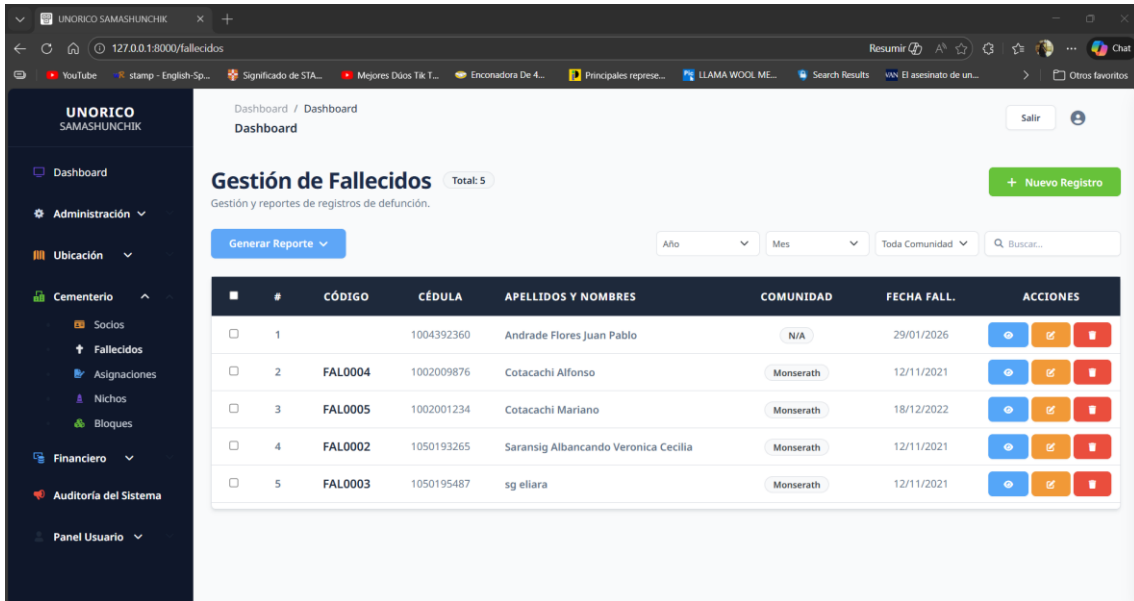


Figura 42 Gestión Fallecidos

Entonces cuando se registra un nuevo fallecido lo que se hace es ingresar su cédula y esperar a ver si carga datos, nos indica un mensaje en verde indicando que ha sido socio en caso de no serlo no se autocompletan los datos y no se muestra el mensaje. Como se muestra a continuación en las Figuras 43-44.

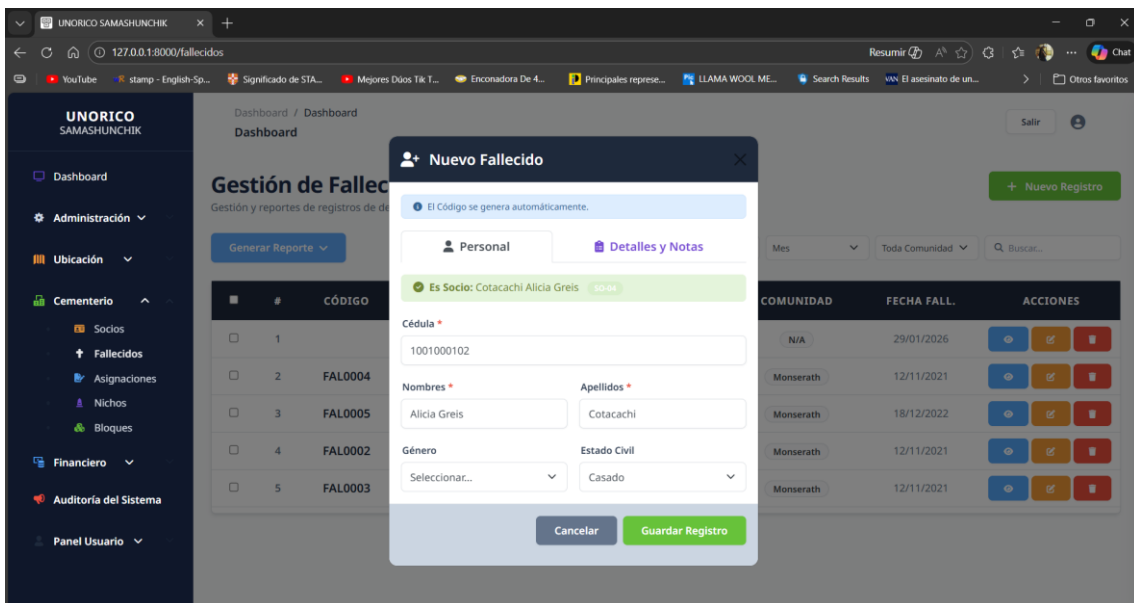
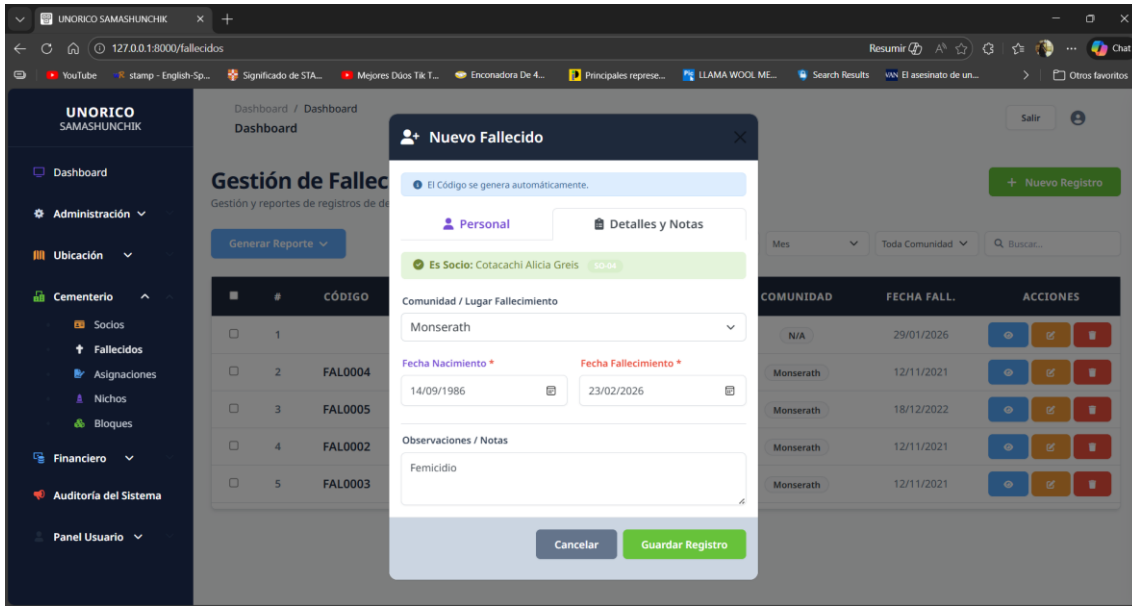
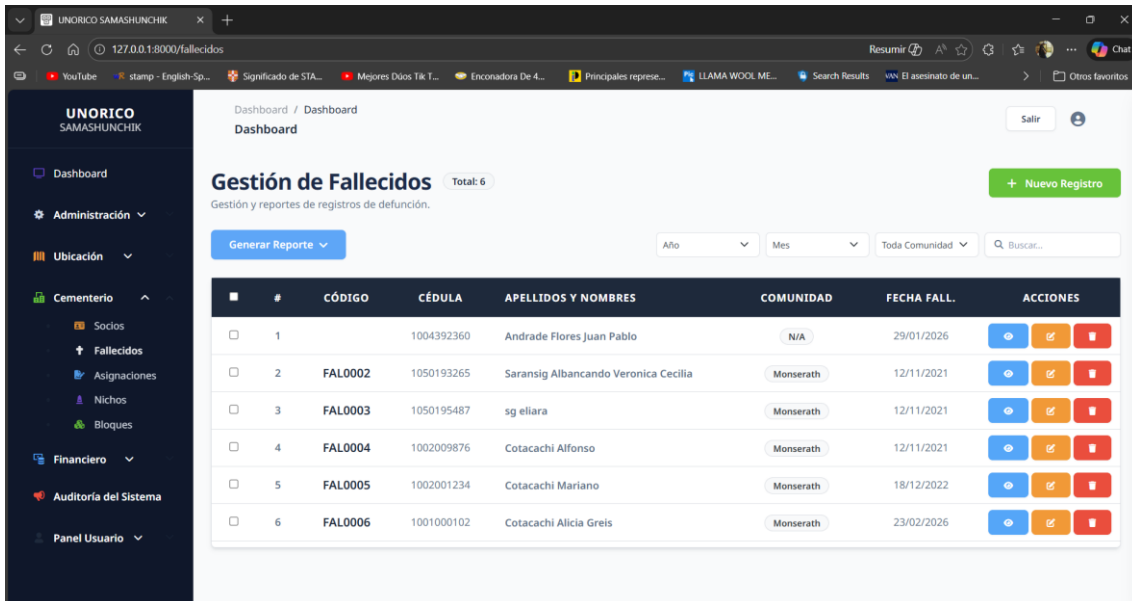


Figura 43 Crear Fallecidos con datos autocompletados pestaña personal

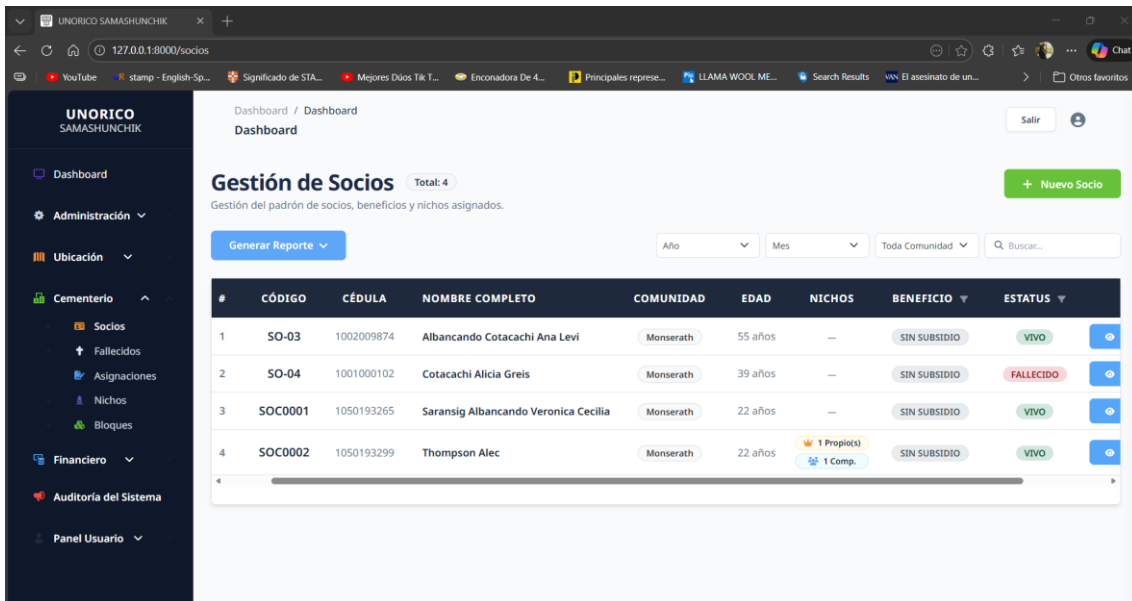


*Figura 44 Crear fallecido con datos autocompletados pestaña detalles y notas*

Una vez creado el registro, se refleja en la tabla de gestión de fallecidos y revisamos la tabla de gestión de socios y el campo “Estatus” debe cambiar de “Vivo” a “Fallecido”. Como está en las Figuras 45-46.

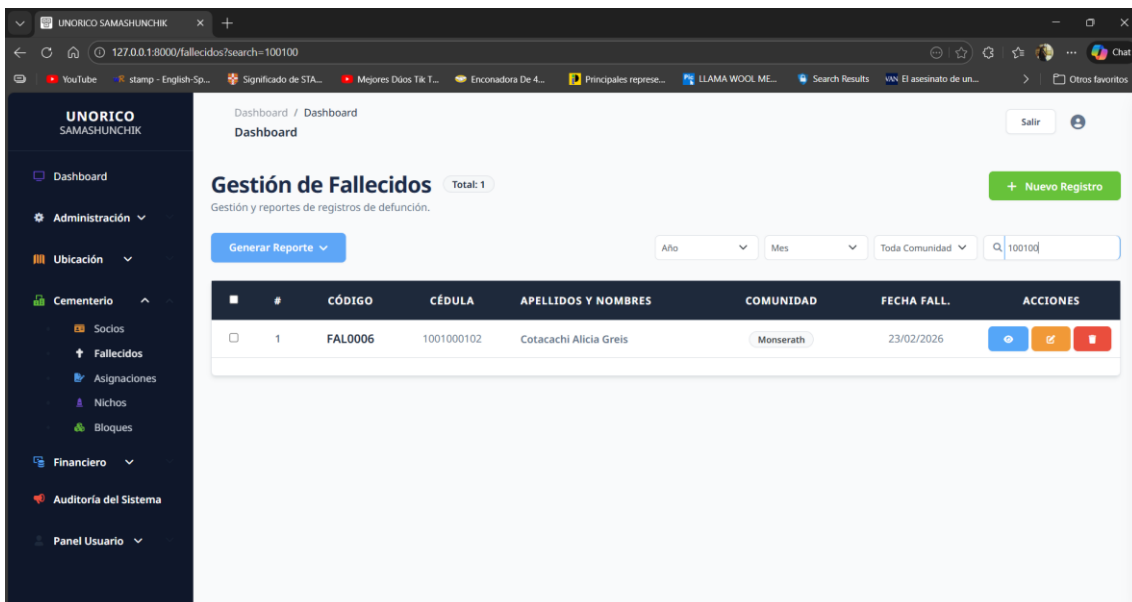


*Figura 45 Registro nuevo fallecido*



*Figura 46 Cambio de estatus a "Fallecido"*

Avanzamos con la HUM-17 (Buscar Fallecidos), ubicada en la Tabla 19, se realizó la búsqueda en tiempo real para localizar a los fallecidos de los socios. Esto se hizo mediante peticiones AJAX que actualizan la tabla de resultados, sin recargar la página y lo busca directamente desde la base de datos. De igual forma, para garantizar la eficiencia de este método se integró un patrón de diseño Debounce de 350 ms y AbortController que cancela automáticamente las solicitudes obsoletas o desactualizadas si el usuario sigue escribiendo. Esto se observa en la Figura 47.



*Figura 47 Buscar fallecido en tiempo real*

Seguimos con la HUM-18 (Inhumación / Asignar) detallada en la Tabla 20, esta sección es el motor central de la aplicación, tiene su respectivo CRUD ya que se relaciona

con tres entidades cruciales: Socio, Nicho, Fallecido. Para un buen control en el backend se programó una regla en la capacidad que puede tener un nicho con límite de 3 fallecidos, tomando en cuenta el incremento del contador de ocupación en caso de llegar al límite, se indica que ese espacio ya está completamente ocupado. La interfaz para registrar una nueva asignación es mediante un modal con tres pestañas: Nicho, Socio y Fallecido. De igual forma tiene sus respectivos filtros y búsqueda. Esto se ilustra en la Figura 48.

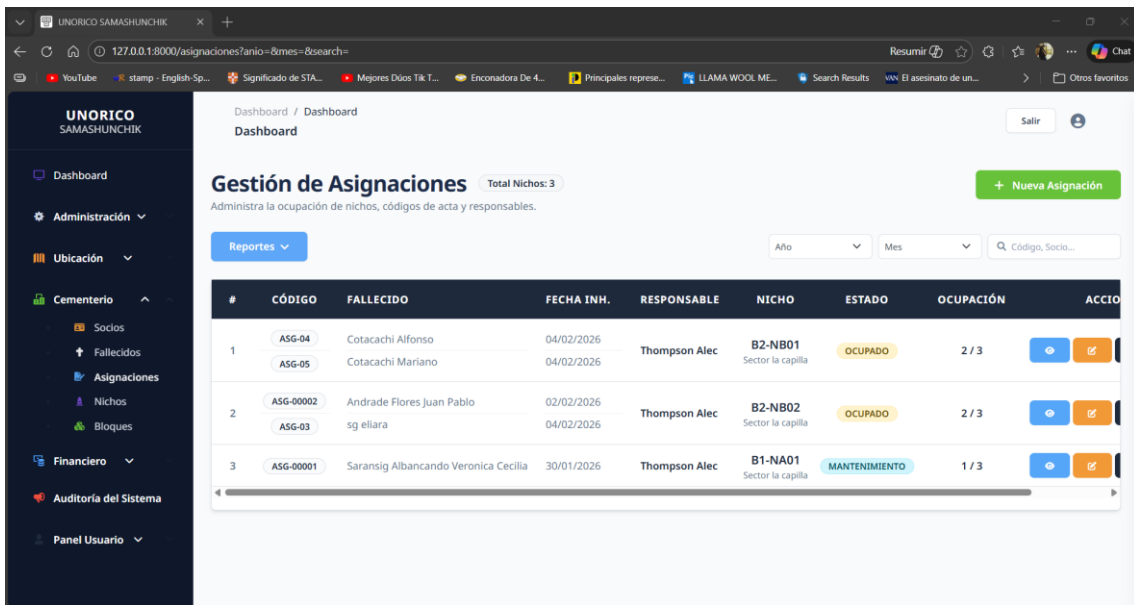
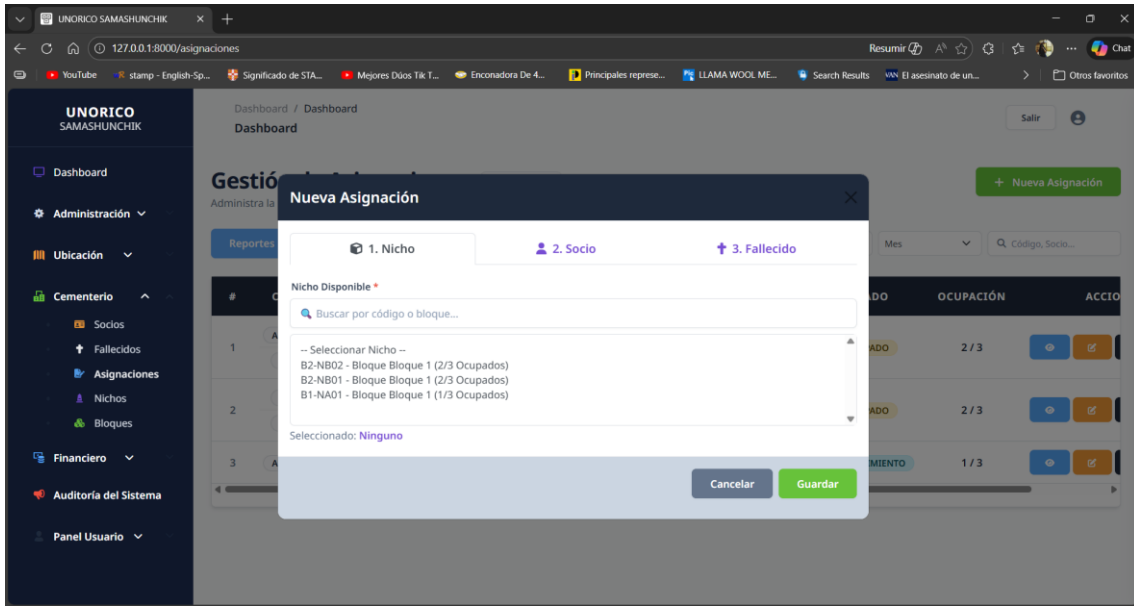
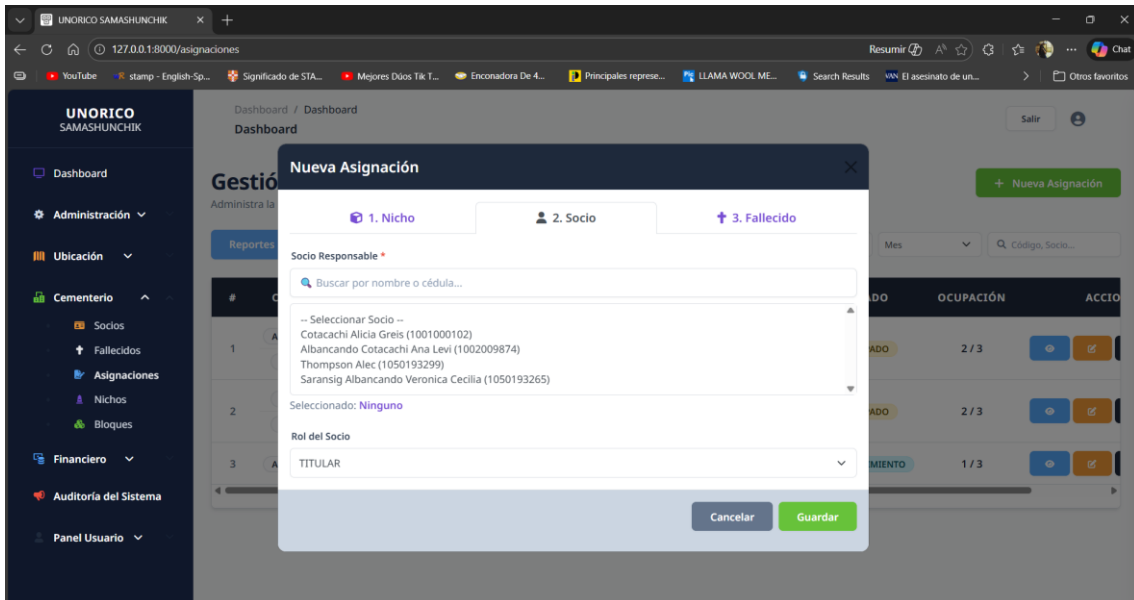


Figura 48 Gestión Asignaciones o Inhumación

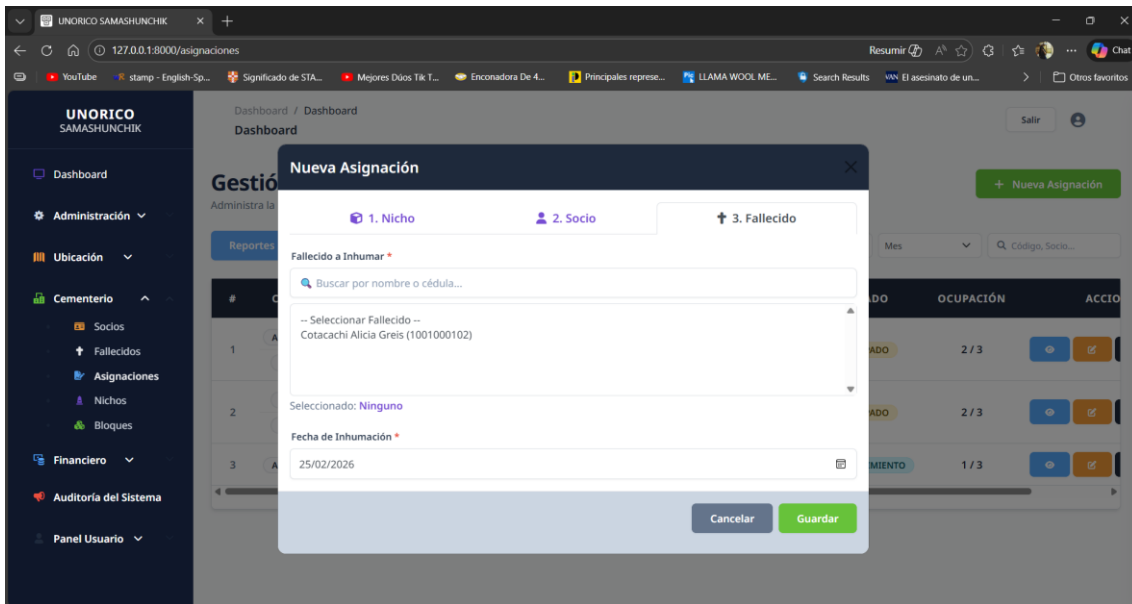
En las Figuras 49-51 se visualiza cómo ingresar una nueva asignación debemos de llenar todos los campos solicitados que están en las diferentes pestañas como: Nicho, Socio, Fallecido, tomándose en cuenta que en la pestaña Nichos solo salen los que siguen con capacidad y en la pestaña Fallecidos solo salen los que aún no fueron asignados a un nicho.



*Figura 49 Crear nueva Asignación pestaña Nicho*

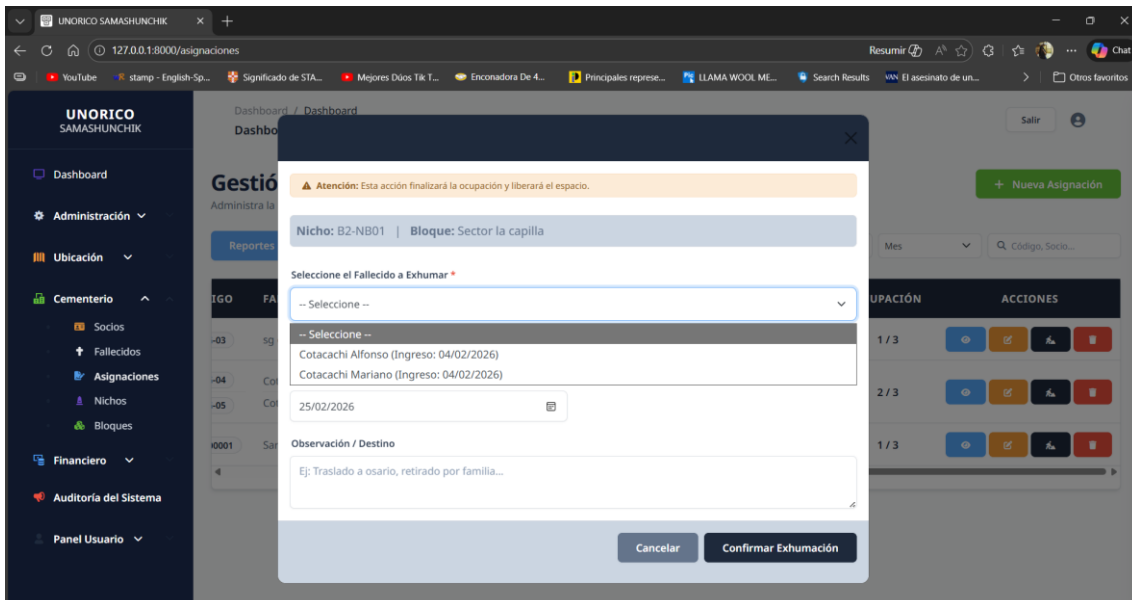


*Figura 50 Crear nueva Asignación pestaña Socio*



*Figura 51 Crear nueva Asignación pestaña Fallecido*

Para la HUM-19 (Exhumación), descrita en la Tabla 21, se realizó la lógica para cuando se requiere retirar los restos. Esto no elimina el registro, solo registra la fecha de salida y se añade una observación (motivo/destino) para lograr tener un historial. También la ocupación de la tumba se actualiza permitiendo la liberación del espacio. Esto se ve mediante un modal en el cual si hay más de un fallecido en el nicho, nos permite seleccionar al fallecido que se va a exhumar y se verifica con una alerta SweetAlert2 para evitar ejecuciones accidentales. Esto se verifica en las Figuras 52-53.



*Figura 52 Exhumar Fallecido*

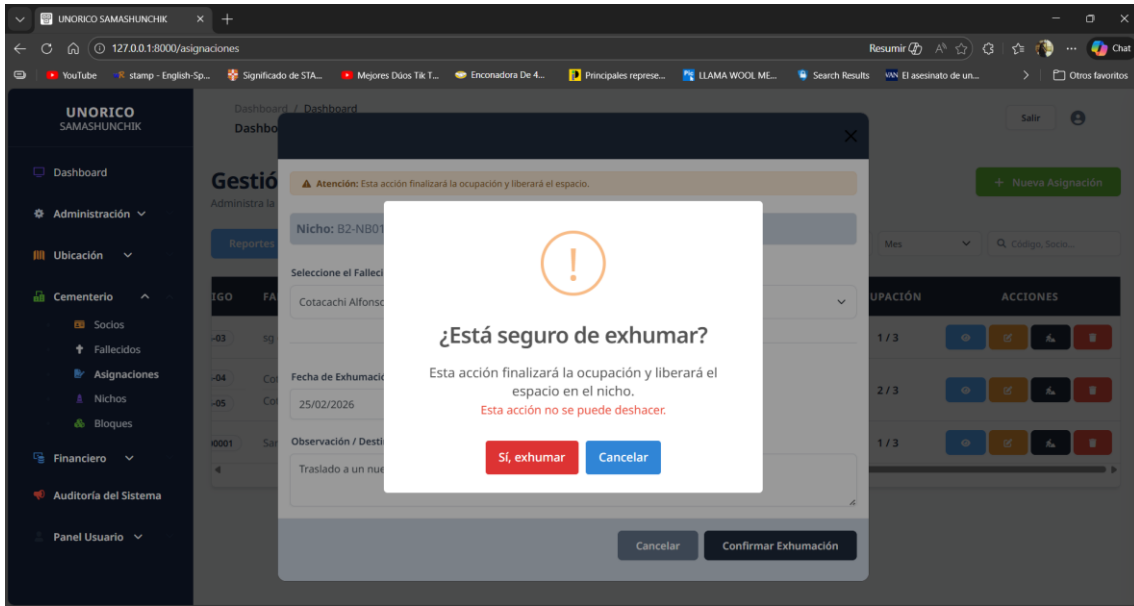


Figura 53 Confirmar Exhumación

Por último, la HUM-20 (Reporte de Asignaciones), correspondiente a la Tabla 22, este apartado nos permite generar documentos en formato PDF, Excel para el reporte general y para solo sacar el reporte de exhumaciones en PDF. Para generar el reporte general también funciona mediante los filtros que se encuentran en la vista principal. Aquello se muestra en la Figura 54-55.

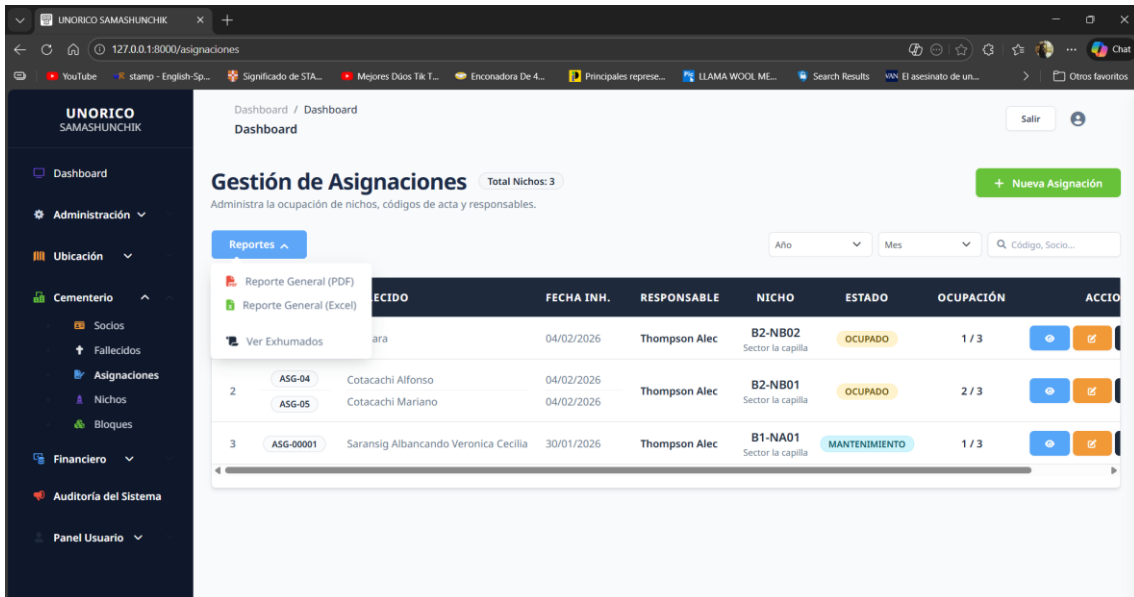


Figura 54 Generar Reportes

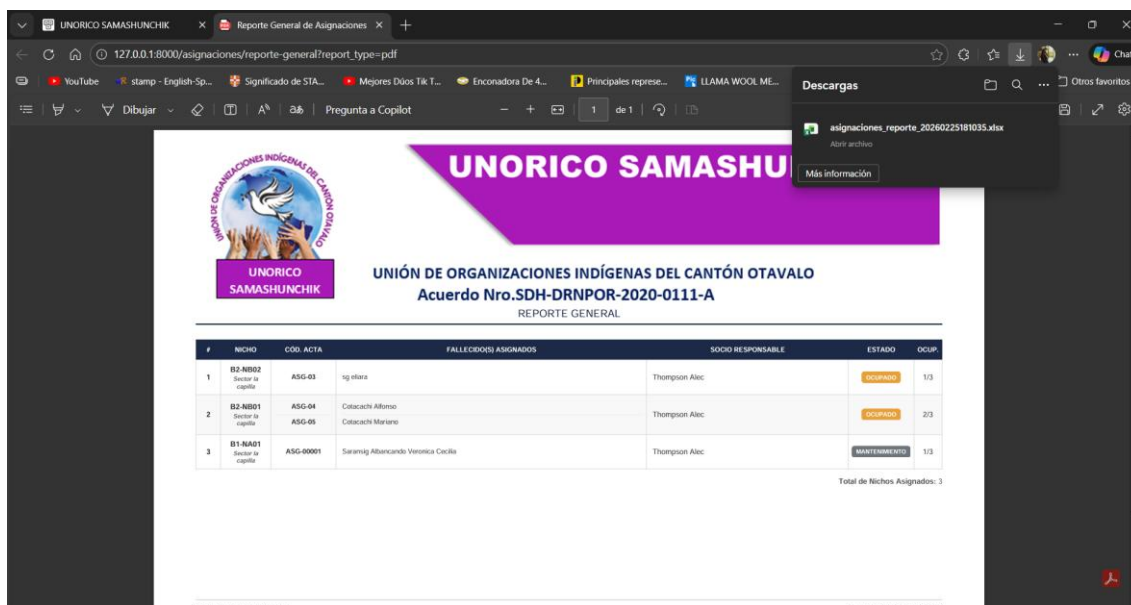


Figura 55 Generar Reporte PDF/Excel

- Revisión – Sprint 3

A continuación, se detalla el cumplimiento de los criterios de aceptación de las Historias de Usuario que fueron elaboradas en el tercer Sprint. Se pueden evidenciar estos criterios en la Tabla 40.

Tabla 40 Revisión - Criterios de Aceptación Sprint 3

Historias de Usuario	Criterio de Aceptación	Cumple
<b>Gestión de Bloques</b>	CA1: La aplicación debe permitir crear un Bloque con los campos: Nombre (obligatorio), Descripción (opcional), Área m <sup>2</sup> (opcional) y polígono del GIS (opcional).	SI
	CA2: Cuando selecciono un polígono GIS el código del bloque se coordina automáticamente con el del GIS y solo se indican los que se encuentren disponibles (Sin asignación).	SI
	CA3: Si no se ingresa el área manualmente pero se ha seleccionado un polígono GIS este lo calcula automáticamente y desde la geometría del mapa.	SI
	CA4: No se puede asignar un polígono GIS que ya está registrado con otro bloque.	SI
	CA5: La aplicación tiene que permitir editar, eliminar bloques.	SI
	CA6: El buscar debe de filtrar por código, nombre del bloque y con su respectiva paginación.	SI

	CA7: Se puede generar reportes en formato PDF y Excel con todos los campos.	SI
	CA1: La aplicación debe permitir crear un nicho con los campos: Bloque (obligatorio), Socio responsable (obligatorio), Tipo (PROPIO/COMPARTIDO, obligatorio), Clase (BÓVEDA/TIERRA, obligatorio), Capacidad (obligatorio, mínimo 1), Estado físico (Bueno, malo, mantenimiento)	SI
	CA2: Cuando se crea un nuevo nicho la ocupación comienza en 0 y la disponibilidad en verdadero.	SI
<b>Gestión de Nichos</b>	CA3: Si se desea se puede vincular con un polígono GIS haciendo que el código se sincronice automáticamente con el código del mapa de igual forma solo se muestran los que no han sido asignados.	SI
	CA4: La aplicación tiene que permitir editar, eliminar nichos.	SI
	CA5: El buscar debe filtrar por nombres, código, cedula y nombres de un socio. También tiene que filtrar por bloque y estado físico.	SI
	CA6: Tiene que generar un código QR para cada uno de los nichos en modo texto (datos del nicho) este se puede previsualizar y descargar en PNG.	SI
	CA7: Debe de generar reportes PDF y Excel con todos sus campos.	SI
	CA1: La aplicación debe de mostrar un mapa interactivo que se visualice los bloques, nichos del cementerio.	SI
	CA2: Al seleccionar o dar clic en un nicho este debe de abrir una ventana mostrando información como: Estado, Bloque, sector, información dentro del nicho.	SI
<b>Ver Mapa Cementerio</b>	CA3: El mapa debe de ser interactivo permitiendo hacer zoom y moverse en el área del terreno.	SI
	CA4: La vista del mapa debe de estar integrada en el apartado asignaciones en la vista detalle donde ya se encuentra la asignación del nicho, socio y fallecido para poder visualizar y que nos lleve directo al nicho del fallecido.	SI
<b>Gestión de Fallecidos</b>	CA1: La aplicación debe permitir registrar a un fallecido con los siguientes campos: Cédula (obligatoria, única), Nombres (obligatorio), Apellidos (obligatorio), Fecha de Nacimiento (obligatorio), Fecha de Fallecimiento (obligatorio), Comunidad (obligatorio), Género (opcional), Estado Civil (opcional) y Observaciones (obligatorio).	SI

	CA2: Cuando se ingresa una cedula que coincide con un socio, se autocompletan los datos automáticamente.	SI
	CA3: Si registra un fallecido y la cedula coincide con la de un socio el estatus de este en la gestión socios cambia automáticamente a “Fallecido”.	SI
	CA4: El código del fallecido se genera automáticamente	SI
	CA5: La aplicación tiene que permitir editar y eliminar registros de fallecidos.	SI
	CA6: Cuando se registra exitosamente a un fallecido debe de indicar un mensaje “Fallecido registrado correctamente”.	SI
<b>Buscar Fallecidos</b>	CA1: La aplicación debe contar con una barra de búsqueda que filtre por cédula, nombres, apellidos o el código de la persona fallecida.	SI
	CA2: La aplicación tiene que permitir filtrar por comunidad por medio de un select desplegable.	SI
	CA3: La aplicación debe permitir filtrar por año y mes de los fallecidos.	SI
	CA4: Los resultados deben de estar con una paginación de 10 registros por página conservando los filtros cuando se cambie de página.	SI
	CA5: El listado en la tabla debe de indicar: código, cédula, nombres, apellidos, comunidad, fecha de nacimiento, fecha de fallecimiento.	SI
<b>Inhumación (Asignar)</b>	CA1: La aplicación debe permitir crear una asignación seleccionando: Nicho (obligatorio) solo se desplegarán nichos disponibles (sin asignación), Socio responsable (obligatorio), Fallecido (obligatorio) tomando en cuenta solo a los socios sin asignación, rol del socio y fecha de inhumación de manera obligatoria.	SI
	CA2: La aplicación tiene que verificar la capacidad del nicho. En caso de estar al límite mostrar un mensaje “EL nicho se encuentra al límite”.	SI
	CA3: El código de asignación se genera de manera automática con el formato ASG-XX	SI
	CA4: Cuando se asigna, la ocupación del nicho se incrementa automáticamente y la disponibilidad se actualiza.	SI
	CA5: La aplicación debe permitir cambiar al socio responsable, fallecido y la fecha de inhumación y su rol.	SI
	CA6: La aplicación tiene que permitir eliminar una asignación equivocada. Entonces la ocupación del nicho se decrementa automáticamente.	SI

	CA7: El listado en la tabla asignaciones debe de mostrar fallecidos, socios, ocupación (1/3), estado del nicho.	SI
	CA8: El buscar tiene que filtrar por código de nicho, nombres, cédula del socio o del fallecido. También se puede filtrar mediante estado del nicho y mes/año de inhumación.	SI
<b>Exhumación</b>	CA1: La aplicación tiene que indicar un formulario para exhumar a un fallecido del nicho seleccionado. No puede exhumar si ya está exhumado solo a los que no lo están.	SI
	CA2: Si el nicho no tiene ocupantes (fallecidos exhumados) debe indicar un mensaje “Este nicho no tiene ocupantes activos (fallecidos sin exhumar)”.	SI
	CA3: Cuando se exhuma se debe de registrar la fecha en la cual se realizó la exhumación (obligatorio) y colocar una observación.	SI
	CA4: Una vez registrada la exhumación la ocupación de nichos decrementa automáticamente y la disponibilidad se recalcula.	SI
	CA5: La aplicación debe permitir generar reportes en PDF y Excel de todas las asignaciones.	SI
	CA6: La aplicación debe permitir generar reporte en formato PDF de todo los exhumados.	SI
<b>Reporte de Asignaciones</b>	CA1: La aplicación generar un reporte en formato PDF y Excel de todas las asignaciones registradas.	SI
	CA2: Tiene que generar los reportes en PDF o Excel con los campos: ID, código, nombre del fallecido, fecha de inhumación, nombre del responsable, nicho, estado, ocupación.	SI
	CA3: El reporte PDF debe de estar en formato A4 horizontal.	SI
	CA4: Cuando se filtre por mes/año el reporte debe de tener un subtítulo indicando que filtros se aplicaron.	SI
	CA5: La aplicación debe permitir generar un reporte en formato PDF solo de exhumaciones.	SI

- Retrospectiva – Sprint 3

En la retrospectiva del Sprint 3, se enfatizó la exitosa implementación de la estructura física y administración operativa, obteniendo la digitalización total del cementerio. Se finalizó la Gestión de Bloques y Nichos, relacionando cada uno de estos con una ubicación geográfica. Un logro destacado fue el desarrollo del Mapa del Cementerio, facilitando una visualización interactiva que ayuda a la administración de la

ubicación de nichos y su disponibilidad en tiempo real. A nivel operativo se realizaron los apartados de Gestión y Búsqueda de Fallecidos, permitiendo un control estricto en los registros de fallecimiento. Los procesos de Inhumación y Exhumación se realizaron de acuerdo con un flujo de trabajo lógico y seguro, asegurando un seguimiento de cada ocupante. El cierre de esta iteración es con Reporte de Asignaciones el cual debe generar documentos en formato PDF y Excel.

### 3.2.5 Sprint 4

- Planificación Sprint 4

Para el cuarto y último sprint se dedicó a realizar el apartado financiero y de seguridad de la aplicación. El propósito esencial fue automatizar el control de deudas y pagos de los socios. También se integró auditoría para registrar y trazar todas las actividades realizadas por parte de los usuarios, asegurando la transparencia e integridad de la información de la organización UNORICO.

*Tabla 41. Sprint Backlog – Sprint 4*

Historia / Código	Nombre	Tarea
HUM-21	Ver Deudas	<p><b>Backend:</b> Consulta que cruza fecha actual vs. último pago y determina los años que se adeudan.</p> <p><b>Frontend:</b> Alerta visual (Badge Roja) en el perfil del socio si tiene deuda pendiente.</p>
HUM-22	Facturación	<p><b>Backend:</b> Generación de orden de cobro vinculando Socio + Servicio + Usuario que cobra.</p> <p><b>Frontend:</b> Interfaz de Caja: Resumen de valores, desglose de IVA (si aplica) y total.</p>
HUM-23	Registrar Pago	<p><b>Backend:</b> Procesar pago, actualizar ultimo_anio_pago del socio y cerrar pago.</p> <p><b>Backend:</b> Controlar los años adeudar, no permitir eliminar registros realizados.</p>
HUM-24	Auditoría (Logs)	<p><b>Backend:</b> Configuración de ActivityLogger para registrar eventos create/update/delete críticos.</p>

---

**Frontend:** Vista de tabla de auditoría filtrable por usuario y fecha (Solo Admin).

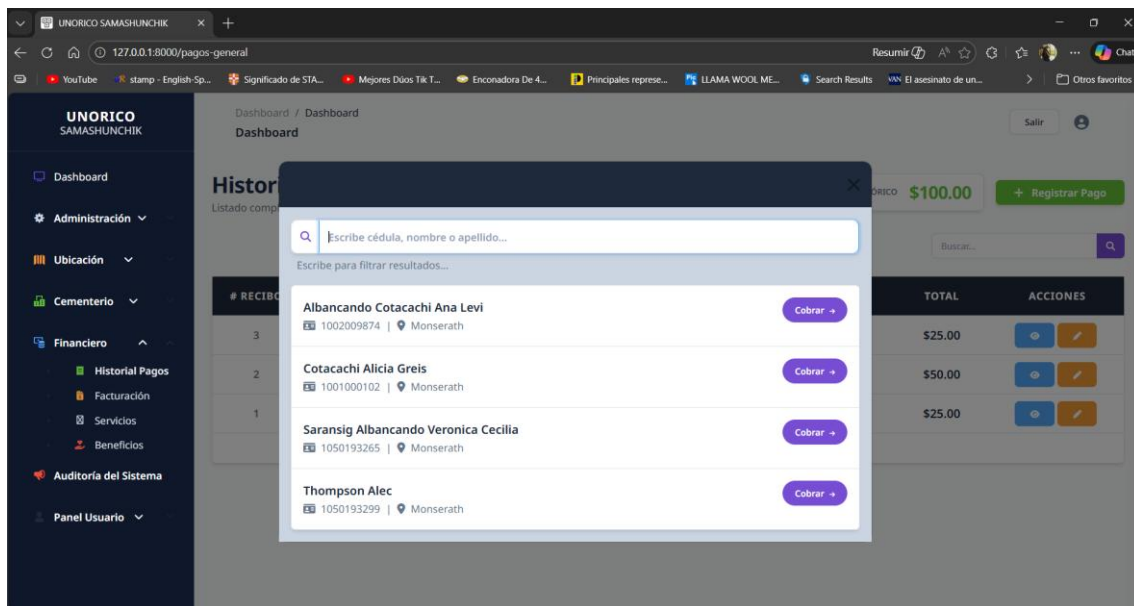
---

En la HUM-21 (Ver Deudas) referenciada en la Tabla 23, se desarrolló la lógica de negocio para la gestión de pagos anuales, en la aplicación se implementó un algoritmo que se encarga de buscar el estado de cuenta de los socios tomando en cuenta el año de inscripción hasta la fecha actual. Tenemos la vista principal para gestionar este historial de pagos. Como se indica en la Figura 56.

# RECIBO	SOCIO	AÑOS CANCELADOS	FECHA	TOTAL	ACCIONES
3	Thompson Alec 1050193299	2026	25/02/2026	\$25.00	
2	Saransig Albancando Veronica Cecilia 1050193265	2020, 2021	25/02/2026	\$50.00	
1	Albancando Cotacachi Ana Levi 1002009874	2026	25/02/2026	\$25.00	

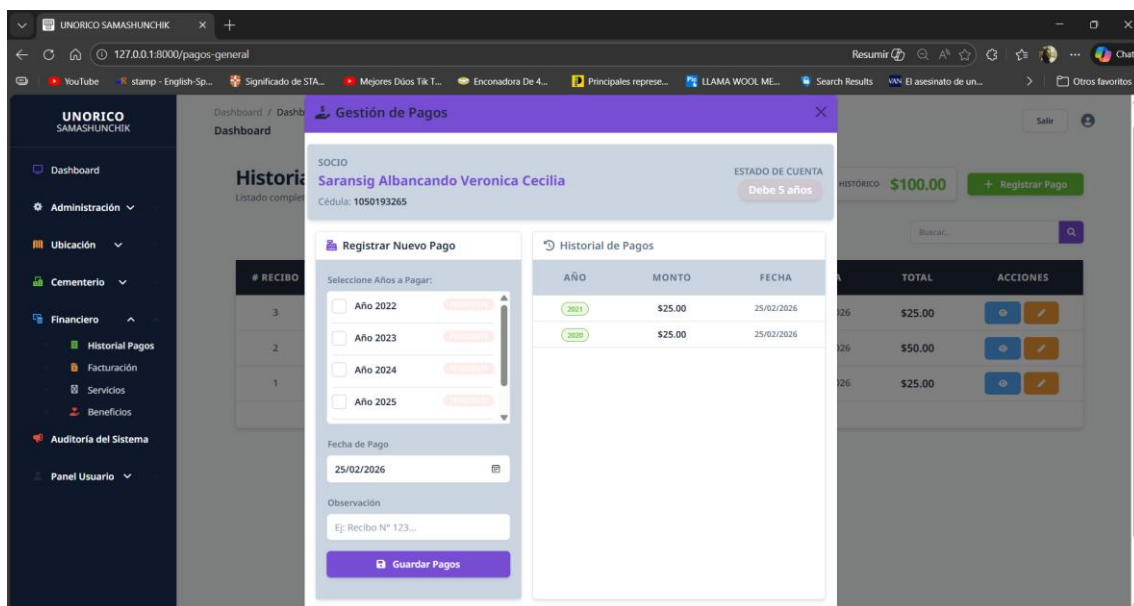
*Figura 56 Gestión Historial de Pagos*

Para verificar los años cancelados o pendientes del socio se da clic en el botón “Registrar Pago” donde nos saldrá una vista con todos los socios que se encuentran registrados en la aplicación, se puede buscar mediante el nombre o cédula y para ver la información de los años pagados y pendientes se da clic en “cobrar”. Esto se muestra en la Figura 57.



*Figura 57 Verificar años cancelados o pendientes*

Una vez seleccionado un socio, se abrirá una ventana la cual nos indica toda la información sobre los pagos del socio, cuántos años debe, cuántos ha pagado y en qué fecha lo ha realizado en un historial de pagos. En la Figura 58 se podrá observar esto.



*Figura 58 Historial de Pago de un Socio*

Para la HUM-22 (Facturación), descrita en la Tabla 24, se buscó estructurar la aplicación con un apartado de cobros para gestionar los ingresos del cementerio. Para el backend se encargó de generar órdenes de cobro, se podía relacionar al socio dinámicamente, los servicios solicitados o también en caso de adquirir un beneficio y con el usuario responsable. Esto se refleja en la Figura 59-61.

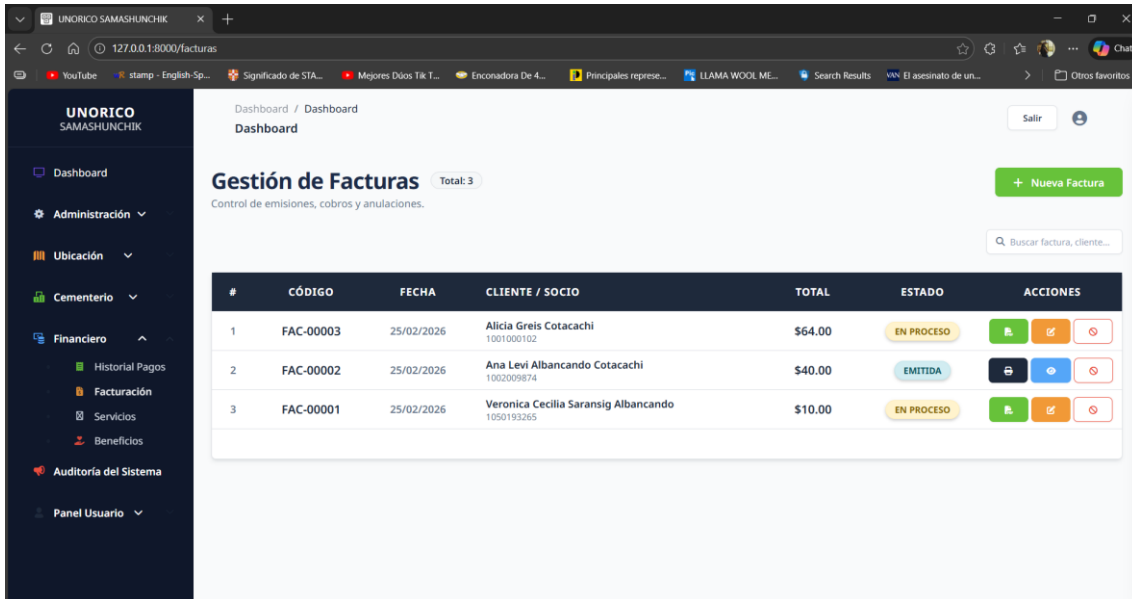


Figura 59 Gestión de Facturas

Primeramente, al generar una factura, inicia como “Borrador”, aquí es donde el encargado define los datos solicitados sin afectar la contabilidad. Pero una vez comprobada la información, se lleva a cabo la “Emisión”, donde la aplicación ya bloquea la edición de esta factura y genera el documento PDF. Como se ve en las Figuras 60-61.

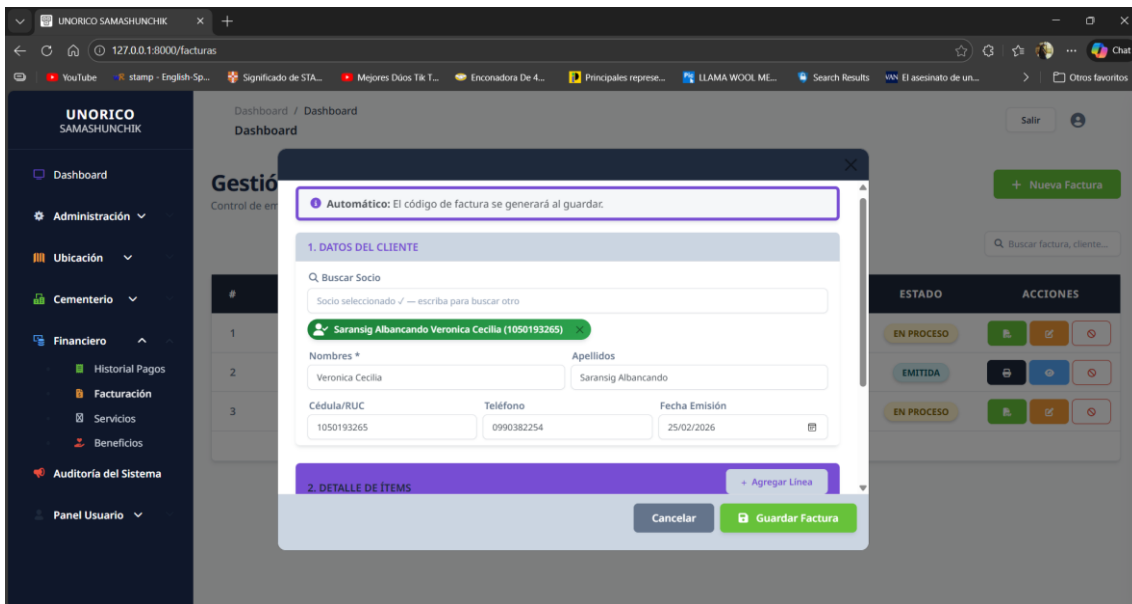
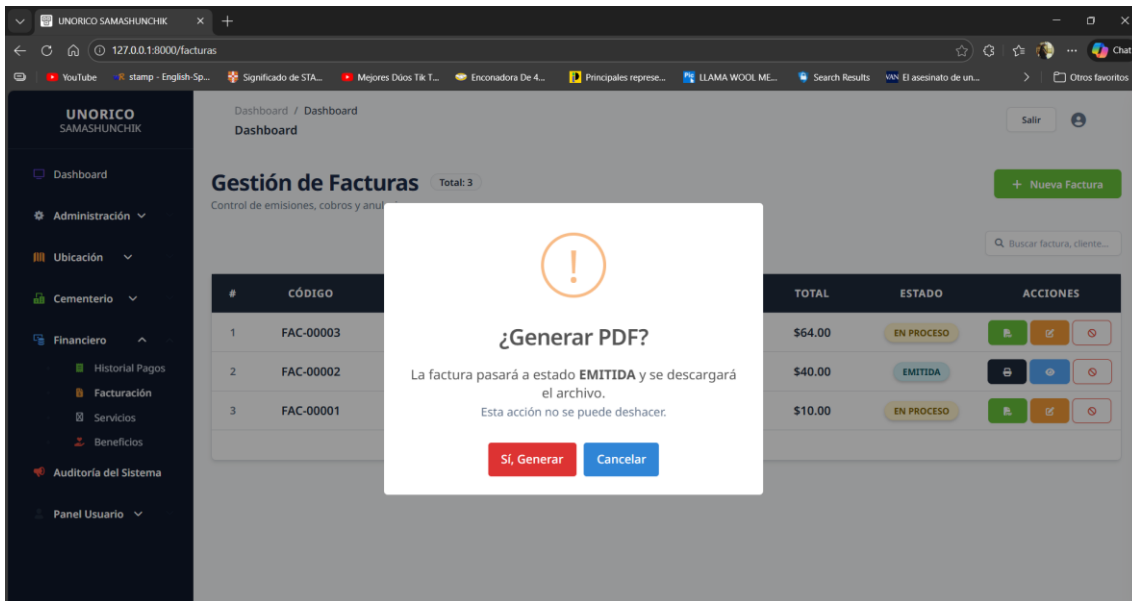
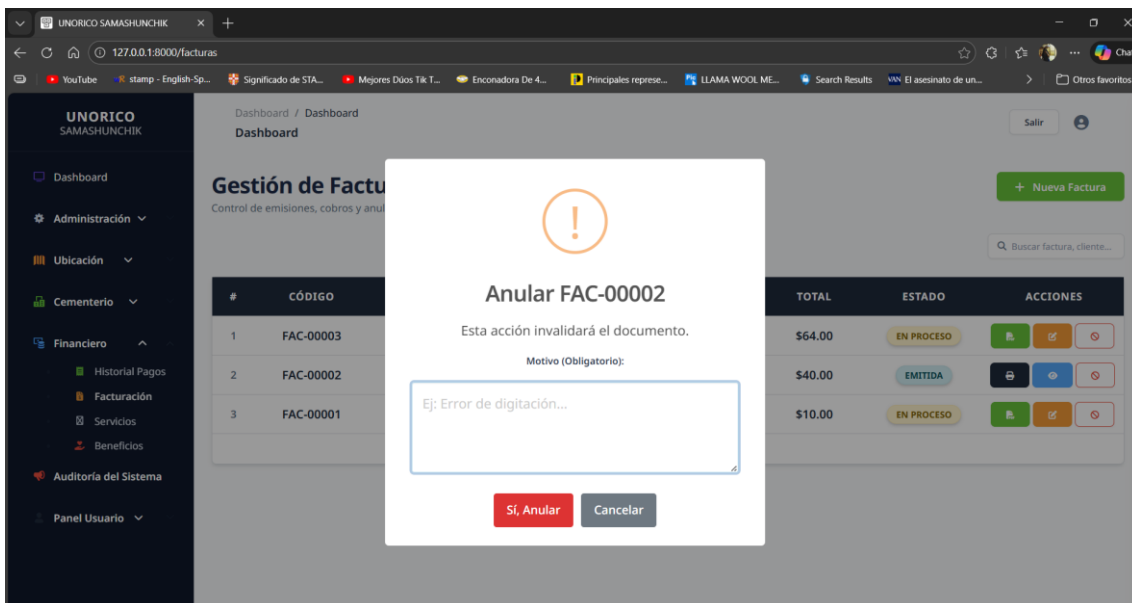


Figura 60 Crear nueva Factura



*Figura 61 Emitir Factura*

Por último, para controlar errores en facturas ya procesadas, se incorporó la función de “Anular”; la aplicación solicita el motivo y ese registro se marca como anulado sin eliminarlo; esto asegura la trazabilidad contable. Esto se muestra en la Figura 62.



*Figura 62 Anular Factura*

En la Figura 63 se puede visualizar cómo se generó la factura en formato PDF una vez que esta fue emitida correctamente.

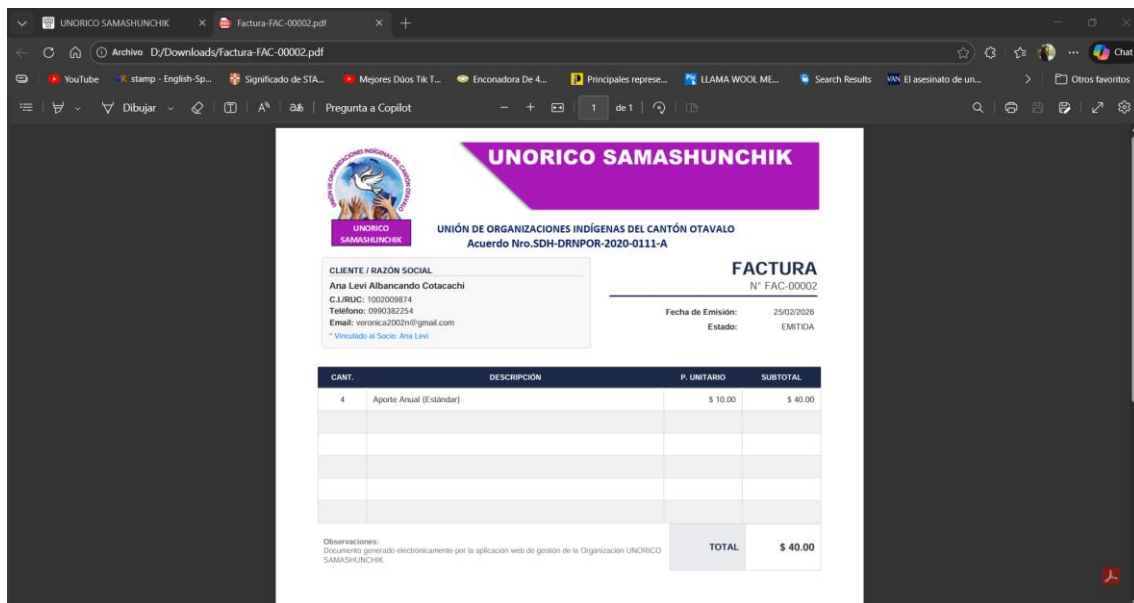


Figura 63 Imprimir Factura

Continuando con la HUM-23 (Registrar Pago) correspondiente a la Tabla 25, se desarrolló esta sección para permitir seleccionar y cobrar múltiples pagos de los años pendientes de los socios; estos ya tienen un valor predeterminado de monto para los pagos anuales, entonces se van calculando automáticamente.

Seleccionamos a un socio y en este caso vemos que no ha pagado ningún año, entonces se van marcando los años que viene a cancelar y se da clic en “Guardar” y se refleja el nuevo registro en la tabla principal con los años cancelados. Esto se indica en las Figuras 64-66.

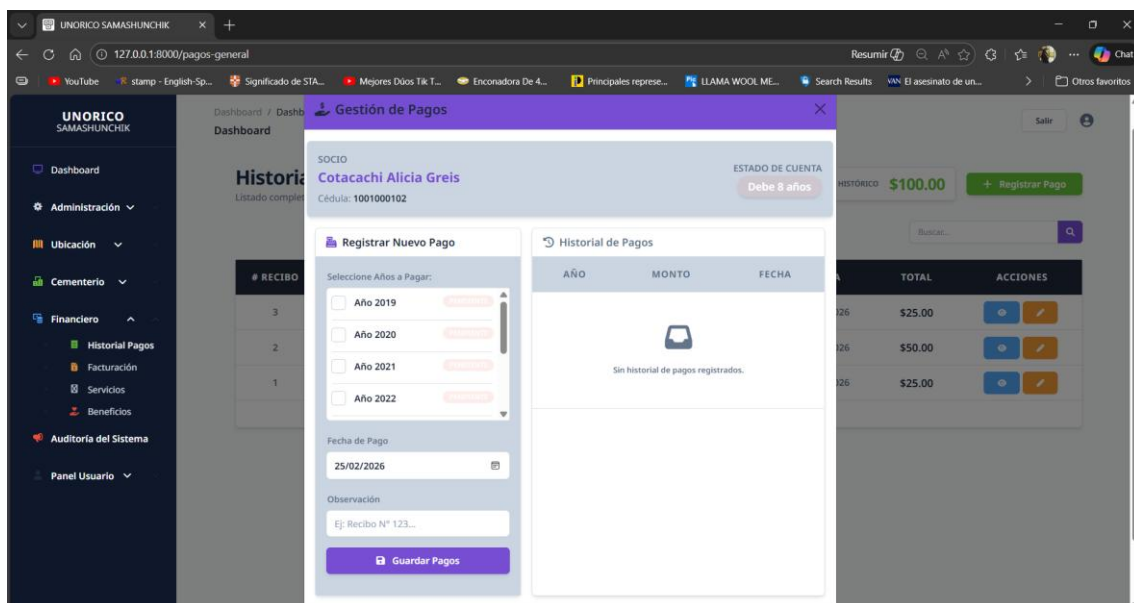


Figura 64 Registrar nuevo pago

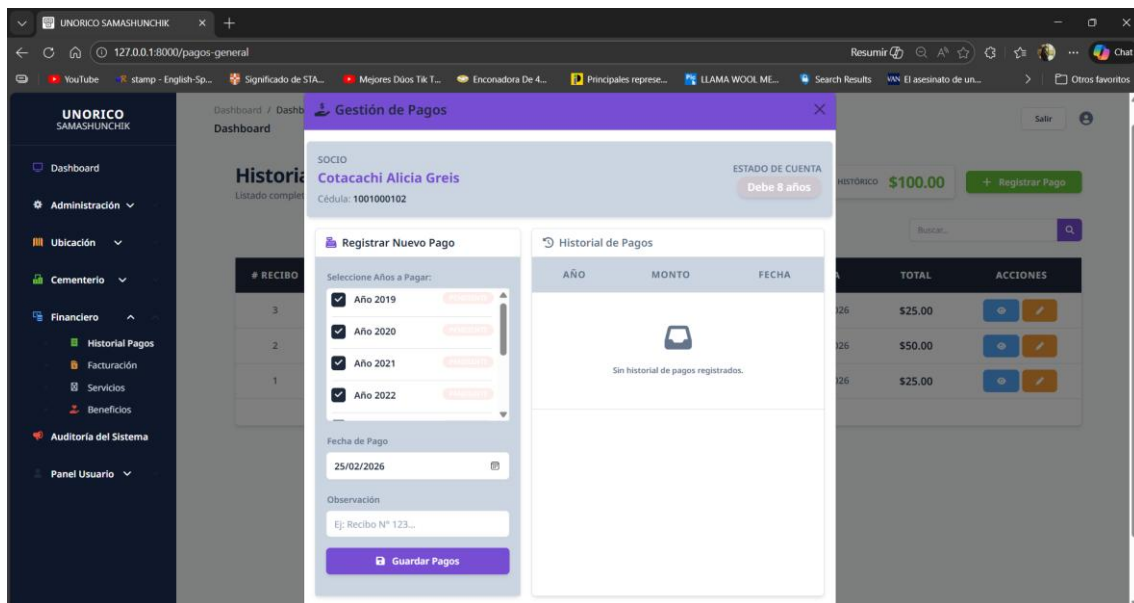


Figura 65 Años seleccionados a cancelar o pagar

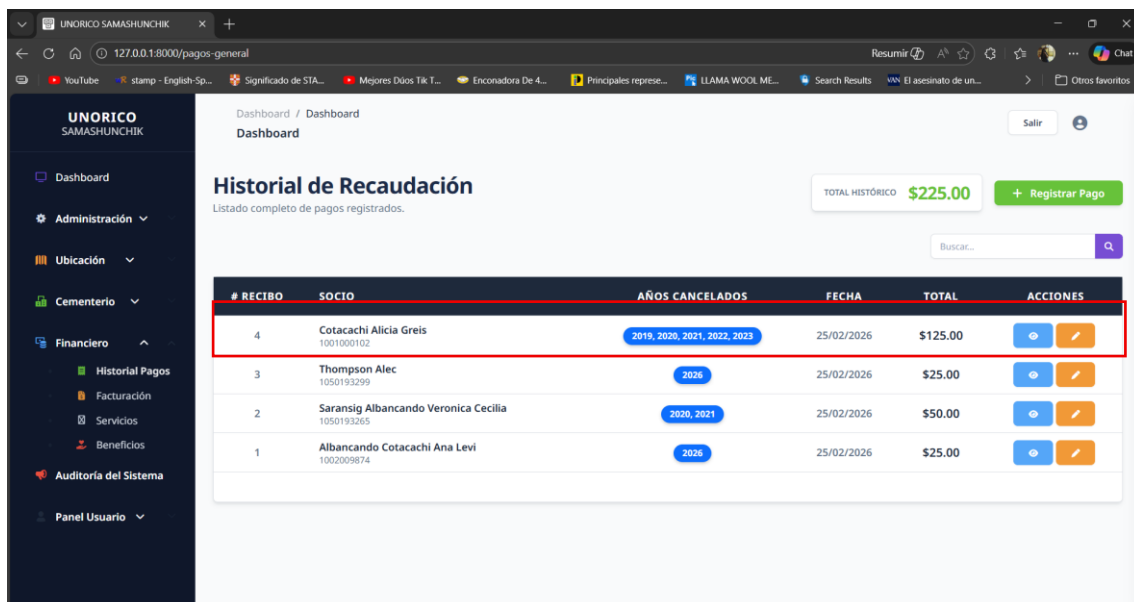


Figura 66 Nuevo registro insertado

Finalmente, la HUM-24 (Auditoría / Logs) detallada en la Tabla 26 fue desarrollada para garantizar el monitoreo total de las actividades realizadas mediante la incorporación del paquete *owen-it/laravel-auditing*. La aplicación obtiene automáticamente los eventos críticos (Crear, Actualizar y Eliminar) de todos los modelos tomando los valores anteriores y los nuevos en formato JSON. Para ver esta vista su ruta está protegida mediante permisos, su interfaz muestra una tabla detallada con búsqueda mediante nombre de usuario y nombre del módulo y también cuenta con sus respectivos filtros como fecha específica o un rango de fechas, lo cual permite realizar seguimiento preciso en los cambios que se ejecutaron. Esto se observa en la Figura 67-68.

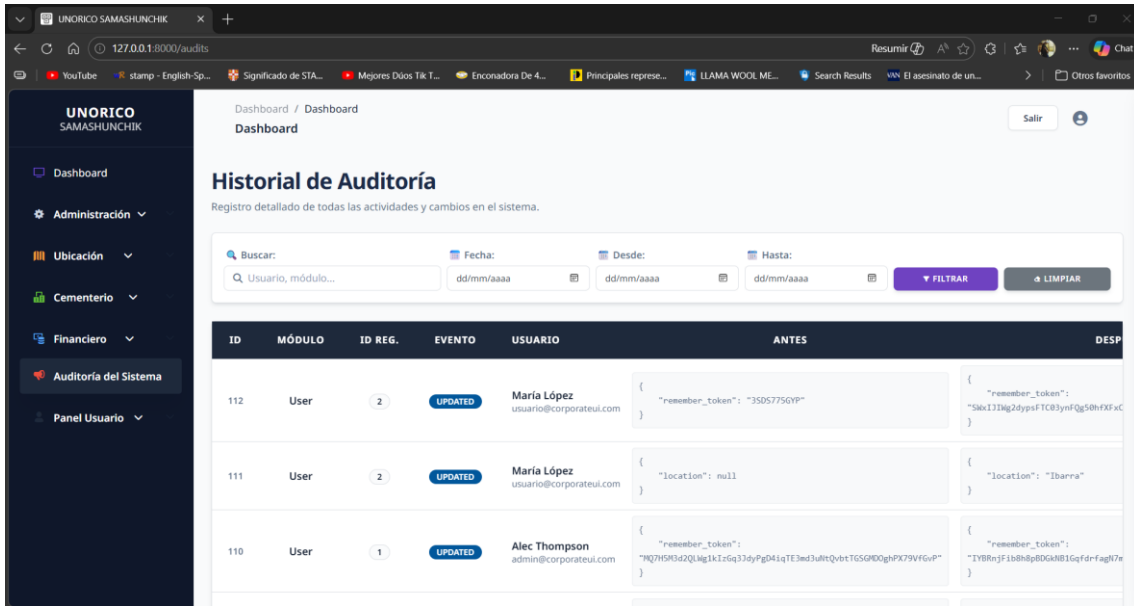


Figura 67 Auditoría

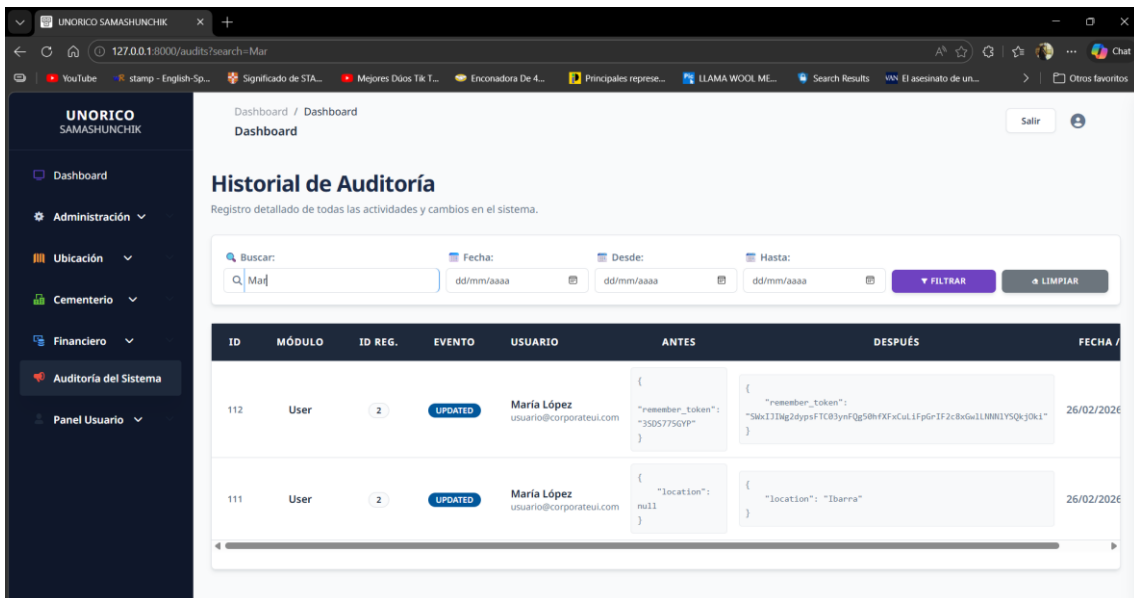


Figura 68 Búsqueda en tiempo real (nombre usuario)

- Revisión – Sprint 4

A continuación, se detalla el cumplimiento de los criterios de aceptación de las Historias de Usuario que fueron desarrolladas en este cuarto último Sprint. Se puede apreciar estos criterios en la Tabla .

Tabla 42 Revisión - Criterios de Aceptación Sprint 4

Historias de Usuario	Criterio de Aceptación	Cumple
----------------------	------------------------	--------

<b>Ver Deudas</b>	CA1: La aplicación debe calcular los años de pago pendientes automáticamente desde la inscripción del socio hasta la fecha actual.	SI
	CA2: En la vista gestión de pagos, se debe de indicar un resumen del socio indicando el estado de la cuenta: en color rojo si tiene años pendientes y en verde que se encuentra al día.	SI
	CA3: En la vista gestión de pagos debe de verse el historial de pagos que ha realizado el socio y el registrar nuevo pago con los años pendientes.	SI
	CA4: La sección de años pendientes solo muestra los años pendientes cuando existen en caso de no tener muestra un mensaje de “No tiene deudas”.	SI
	CA5: Al dar clic en el botón show se abrirá una ventana para ver los detalles del historial de pago del socio detallando años pagados, pendientes y su información.	SI
	CA6: La venta de historial de pagos debe de tener un botón “Ir a cobrar” en cada socio para dirigirse al cobro.	SI
	CA7: Para acceder al historial del socio debe de estar protegido por permisos.	SI
	CA8: Para crear un pago y ver deben de estar con permisos para acceder o ejecutar estas acciones.	SI
	CA9: La lista debe estar ordenada cronológicamente por fecha de fallecimiento (del más reciente al más antiguo) o alfabéticamente.	SI
<b>Facturación</b>	CA1: La aplicación debe permitir crear una factura escogiendo un socio registrado por medio del buscar en tiempo real ya se cedula o nombres y apellidos de igual manera si o es un socio se puede ingresar los datos de un cliente externo de forma manual.	SI
	CA2: Cuando seleccionamos un socio registrado los campos se llenan automáticamente con los datos de él.	SI
	CA3: El campo nombre y apellidos son obligatorio para el cliente, los demás campos pueden ser opcionales.	SI
	CA4: Cuando se realice la factura obligatoriamente debe estar la fecha de emisión y por defecto se establece la fecha actual.	SI
	CA5: La aplicación tiene que permitir agregar múltiples líneas de detalles. Cada una de las líneas requiere seleccionar un Tipo que puede varias entre Beneficio o Servicio, después debe de seleccionar un ítem del catálogo correspondiente, cantidad y precio unitario.	SI
	CA6: Cuando se selecciona un ítem del catálogo, el precio de este debe llenarse automáticamente con el valor predeterminado ya sea el beneficio o servicio.	SI

CA7: El subtotal de cada línea y el total deben se calcularse en tiempo real.	SI
CA8: La factura por lo menos debe tener un ítem en detalle si no hay mostrar un mensaje “Debe agregar al menos un ítem”.	SI
CA9: El código único debe generarse automáticamente en formato FAC-00001 de manera consecutiva.	SI
CA10: Cuando se guarda la factura exitosamente se crea en el estado de “borrador” y muestra un mensaje “Factura FAC-xxxx guardada como borrador” con esto puede editar o emitirla.	SI
CA11: Las facturas que se encuentran estado de borrador se puede editar. Si la factura ya está emitida o anulada ya no permite editar.	SI
CA12: Cuando se edita se debe de poder modificar los datos del cliente, su fecha y los ítems del detalle (agregar, retirar, cambiar cantidades).	SI
CA13: Cuando se actualizan los campos y ítems se cambian completamente por los nuevos datos. El total y subtotal deben de recalcularse.	SI
CA14: Al actualizar exitosamente se debe indicar un mensaje “Factura actualizada correctamente”.	SI
CA15: Cuando se presione el botón para “Generar PDF”, la aplicación lanza una alerta de confirmación (SweetAlert) indicando que la factura pasara a estado “Emitida” y no se puede deshacer.	SI
CA16: Cuando se confirma la emisión de la factura, se registra la fecha de emisión y el usuario que lo hizo y se descarga automáticamente en archivo PDF.	SI
CA17: Una factura ya emitida no se puede editar. Solo permite imprimir el PDF, ver los detalles o anular.	SI
CA18: Se puede anular una factura si está en el estado de borrador o emitida pero no se puede anular si ya fue anulada.	SI
CA19: Cuando se anula se registra la fecha de anulación, el usuario que lo hizo y el motivo. Se muestra el mensaje “La FAC-XXXX ha sido anulada”.	SI
CA20: Una vez anulada la factura no se puede editar ni emitir. Solo permite ver el motivo de anulación por el botón “Motivo” que se muestra mediante un cuadro de información.	SI
CA21: Las facturas no se puede eliminar por políticas de seguridad y auditoría.	SI
CA22: En la tabla de la gestión de Facturas se debe de ver el listado el cual cuenta con las columnas: número de fila, código, fecha, cliente/socio (nombre + cédula), total y	SI

	estado de color (amarillo = EN PROCESO, celeste = EMITIDA, rojo = ANULADA).	
	CA23: La aplicación debe de tener una barra de buscar en tiempo real mediante código de factura, nombres del socio o cliente y cédula.	SI
	CA24: El acceso a este apartado debe de estar controlado mediante roles y permisos.	SI
	CA1: La aplicación cuando se va a registrar un pago debe de abrir una ventana para buscar al socio.	SI
	CA2: El buscar debe de filtrar a los socios en tiempo real por medio de nombres, apellidos y cédula.	SI
	CA3: Una vez seleccionado al socio debe de dar clic en él o n un botón de cobrar que redirija a la ventana gestión de pagos.	SI
	CA4: En caso de no existir un socio con ciertos datos de búsqueda debe de salir “No se encontró ningún socio con esos datos”	SI
	CA5: Cuando estamos en la vista gestión de pagos del socio debe de haber dos secciones una para los años pendientes y el otro historial de los pagos que haya realizado	SI
	CA6: La cabecera de la vista gestión pagos debe de mostrar los datos del socio y como está su estado.	SI
<b>Registrar Pago</b>	CA7: En la parte donde se encuentran los años pendientes deben de estar con checkboxes individuales. Donde el usuario debe seleccionar los años que debe de cobrar.	SI
	CA8: En caso de que el socio no tenga años pendientes en esa sección sale un cuadro con el mensaje “No tiene deudas pendientes”.	SI
	CA9: El campo fecha pago es obligatorio y por defecto se establece la fecha actual.	SI
	CA10: El campo observación es opcional lo que permite agregar una nota.	SI
	CA11: Debe se seleccionar un año pendiente para guardar el pago caso contrario indica un mensaje “Debe de seleccionar al menos un año para guardar el pago”.	SI
	CA12: Cuando se va a guardar un pago la aplicación verifica si existe un recibo del socio en la misma fecha (en el mismo día). En caso de no existir se crea un nuevo registro y se relaciona los pagos relacionados y si existe se agregan los nuevos años al registro existente.	SI

	CA13: No se permite registrar un año que ya fue cancelado.	SI
	CA14: El total del historial de pagos se recalcula automáticamente sumando los nuevos pagos realizados (suma de todos los registros).	SI
	CA15: Cuando se guarda correctamente, debe mostrar un mensaje “Pago registrado correctamente”.	SI
	CA16: La aplicación debe indicar un listado con paginación de 10 por página.	SI
	CA17: El listado de la tabla debe contar con las siguientes columnas: # Recibo, Socio (nombre + cédula), Años Cancelados (de color azul con los años), Fecha, Total y Acciones.	SI
	CA18: El buscar debe de filtrar por cédula, nombre y apellido del socio	SI
	CA19: La aplicación permite corregir los años que fueron asignados a un registro.	SI
	CA20: Solo se indican los años que son menores al año actual (no permite años futuros).	SI
	CA21: Se puede modificar el campo de observación del registro.	SI
	CA22: Cuando se guarda el registro editado, se corrige los pagos realizados y se crean los nuevos pagos seleccionados, se recalcula el total del registro y muestra un mensaje “Registro Actualizado”.	SI
	CA23: La aplicación no debe de eliminar los registros que ya han sido creados.	SI
	CA24: El acceso para este apartado debe de estar controlado por roles y permisos para ejecutar todas las operaciones que se realiza aquí.	SI
<b>Auditoría (Logs)</b>	CA1: La aplicación debe registrar automáticamente las acciones realizadas por el usuario cada vez que crea, edita o elimina información en cualquier parte de la aplicación.	SI
	CA2: Cada acción realizada, se debe de obtener obligatoriamente: el usuario que la realizo, la acción efectuada, a que modulo afecto, el dato que afecto y la fecha que fue hecho.	SI

CA3: En caso de ediciones, la aplicación debe indicar que información se tenía antes y como quedo después del cambio.	SI
CA4: Los registros deben de ser claros y sencillos de leer para el administrador o auditor.	SI
CA5: La aplicación tiene que brindar una pantalla donde se indique el Historial de Auditoria en la cual se listen todas las actividades realizadas en orden cronológico.	SI
CA6: Debe de permitir la búsqueda por nombre de usuario y modulo y filtrado por fechas un día específico o en un rango de fechas.	SI
CA7: Los registros de auditoría no pueden ser editados ni eliminados por ningún usuario.	SI
CA8: Para acceder a este apartado se debe de tener roles y permisos.	SI

- Retrospectiva -Sprint 4

En la retrospectiva del Sprint 4, se destacó la finalización del entorno financiero y del control del sistema (auditoría). Se implementó un potente apartado de facturación y registro de pagos, posibilitando la emisión de comprobantes y gestión de ingresos de modo automatizado. Un logro clave fue la unificación del apartado de Visualización de Deudas, lo que facilita identificar los años pendientes de los socios de forma rápida, mejorando considerablemente la gestión de pagos anuales. Finalmente, la integración del apartado de Auditoría se encarga de que cada acción ejecutada en la aplicación permanezca registrada con seguimiento estricto, cumpliendo con las normas de seguridad y transparencia. Al finalizar estos apartados se consiguió una sincronización entre la gestión financiera y operativa, dejando la aplicación lista para producción.

### 3.3 Desarrollo Etapa Pos-Juego

En esta etapa se llevó a cabo el despliegue de la aplicación web para la gestión del cementerio de Otavalo incorporando todos los apartados desarrollados (seguridad, ubicación, socios, fallecidos, infraestructura y finanzas) en una infraestructura tecnológica moderna y adaptable. Este procedimiento permitió unificar un flujo de trabajo de principio a fin, garantizando que la aplicación sea apta para administrar todo el ciclo de vida de la gestión del cementerio en un entorno real.

### **3.3.1 Despliegue del Producto**

La aplicación principal (backend y frontend) fue alojada en Render, una plataforma la cual brinda servicios en la nube configurada para el despliegue de aplicaciones web. El backend fue desarrollado en Laravel; este fue dockerizado por medio del uso de un contenedor especializado (richarvey/nginx-php-fpm), lo que permitió separar su entorno de ejecución, gestionar rigurosamente las dependencias y garantizar una configuración estándar entre los entornos de desarrollo y producción. Esta estrategia basada en contenedores redujo significativamente el despliegue y mitigó los riesgos de incompatibilidad entre servidores.

La preservación de los datos se gestiona por medio de Supabase, aprovechando su motor de base de datos PostgreSQL de óptimo rendimiento. Al contrario del entorno local, esta arquitectura independiente permite que la base de datos sea autosuficiente del servidor de la aplicación, asegurando mayor seguridad. Las solicitudes son gestionadas de forma segura (HTTPS), protegiendo la integridad y confidencialidad de todas las peticiones administrativas y financieras.

El despliegue que se llevó a cabo en Render incorpora un proceso de integración y Despliegue Continuo que se vincula desde el repositorio del código fuente. Cada modificación se gestiona automáticamente para generar una nueva imagen Docker y las actividades principales de configuración de Laravel, como la optimización de Autoloader y administración de permisos en Storage y cache. Esto permitió la operación constante de la aplicación web sin interrupciones.

### **3.3.2 Entrega del Producto**

Para la entrega de la aplicación web desarrollada, se llevaron a cabo todos los requerimientos definidos por la directiva de la Organización UNORICO SAMASHUNCHIK , incluyendo la elaboración de la documentación del manual de usuario. Igualmente se llevó a cabo una reunión de revisión en la que se analizó tanto el funcionamiento de la aplicación como la calidad del documento proporcionado. Al final, se consolidó el proceso por medio de un acta de entrega, la cual certifica que la aplicación fue aceptada de manera satisfactoria por la organización. El certificado ser visto en el Anexo E.

## **CAPÍTULO III**

### **RESULTADOS Y ANÁLISIS**

#### **4.1 Herramientas de Evaluación**

##### **4.1.1 Modelo de evaluación seleccionado**

Para determinar la efectividad de la aplicación web de gestión del cementerio, se ha seleccionado el Modelo de Éxito de Sistemas de Información de DeLone & McLean. Este modelo es el estándar idóneo para su evaluación, ya que valida cómo el sistema gestiona información crítica y mejora los procesos administrativos. Investigaciones recientes del año 2025, como la de Firdaus y Adrianto [62], confirman la vigencia de este modelo para medir la calidad de software en instituciones que manejan datos financieros y de registro, tal como lo hace el sistema propuesto.

##### **4.1.2 Delimitación de la evaluación**

El alcance de la evaluación se circunscribe a la validación funcional y operativa de la "Aplicación web de Gestión de Cementerio" en un entorno de producción real. El estudio se delimitó a la participación del personal administrativo y operativo de la organización, que interactúan directamente con los módulos de registro de socios, fallecidos, nichos, bloques, servicios, beneficios, control en recaudación (pagos).

Para garantizar la pertinencia del estudio, la evaluación se centra en medir las dimensiones de éxito propuestas por el modelo de DeLone & McLean, utilizando como referencia metodológica los criterios de validación establecidos por Ojo [63]. De esta manera, el análisis se enfoca estrictamente en determinar la aceptación tecnológica y la mejora de los procesos administrativos dentro de la organización, excluyendo aspectos de desarrollo de código o infraestructura de servidores ajenos al uso del software.

##### **4.1.3 Matriz Operacional**

Para la elaboración del cuestionario, se tomó como base el instrumento validado en el artículo científico "Validation of the DeLone and McLean Information Systems Success Model" de Ojo [63]. Las preguntas originales de este estudio fueron adaptadas y contextualizadas para responder a la realidad de la "Aplicación web para la Gestión de Cementerio", asegurando así el rigor científico en la medición.

Para la recolección de datos, se diseñó un cuestionario de 22 ítems distribuidos en las seis dimensiones del modelo. La redacción de cada pregunta se fundamenta en los indicadores validados por Ojo [63], adaptando el contexto hospitalario del estudio original a las necesidades de gestión de información de nichos, socios, fallecidos y servicios. Esto se ilustra en la Tabla 43.

La escala de medición será:

- 1: Totalmente en desacuerdo
- 2: En desacuerdo
- 3: Neutral
- 4: De acuerdo
- 5: Totalmente de acuerdo

*Tabla 43 Encuesta con el Modelo DeLone & McLean aplicada en la evaluación de la aplicación web*

<b>Dimensión</b>	<b>Ítem (pregunta)</b>
Calidad del Sistema	<ol style="list-style-type: none"> <li>1. La aplicación web es fácil de aprender y utilizar para nuevos usuarios.</li> <li>2. La aplicación carga rápidamente al realizar búsquedas de socios, fallecidos o nichos.</li> <li>3. La aplicación funciona de manera estable y no presenta errores frecuentes (cierres inesperados).</li> <li>4. La interfaz y los menús del sistema son claros y fáciles de comprender.</li> </ol>
Calidad de la Información	<ol style="list-style-type: none"> <li>5. La información mostrada sobre los: socios, pagos, nichos y fallecidos es precisa y se alinea con la realidad.</li> <li>6. La aplicación web brinda información completa y necesaria para realizar los procesos requeridos.</li> <li>7. Los datos y los reportes se muestran en un formato comprensible y fácil de leer.</li> <li>8. La información se encuentra constantemente actualizada al instante de consultar.</li> </ol>

Calidad del Servicio	<p>9. La aplicación web muestra mensajes de ayuda o validación claros cuando ocurre o comete un error.</p> <p>10. Cuenta con la ayuda o la orientación necesaria para solucionar dudas sobre el uso de la aplicación web.</p> <p>11. La capacitación o indicaciones recibidas son suficientes para manejar la aplicación correctamente.</p>
Uso / Intención de Uso	<p>12. Uso (o usaré) la aplicación habitualmente para mis actividades en la administración del cementerio.</p> <p>13. Se apoyo en la aplicación para llevar a cabo consultas rápidas y evitar la búsqueda manual.</p> <p>14. Tiene la intención de continuar usando esta aplicación en el futuro para sus funciones administrativas.</p>
Satisfacción del Usuario	<p>15. Está satisfecho con el rendimiento de la aplicación en comparación con las operaciones manuales anteriores.</p> <p>16. La aplicación satisface con todas sus expectativas para la gestión del cementerio.</p> <p>17. Se siente cómodo y seguro interactuando con esta aplicación web.</p>
Beneficios Netos	<p>18. La aplicación le ayuda a reducir el tiempo en la búsqueda y registro de información.</p> <p>19. El uso de la aplicación disminuye considerablemente el consumo de papel y archivos físicos.</p> <p>20. La aplicación mejora la transparencia, la gestión de pagos, nichos, fallecidos y socios.</p> <p>21. La aplicación ayuda a prevenir equivocaciones o fallos como la duplicidad de registros o pérdida de información.</p> <p>22. El código identificador facilita el proceso de búsqueda en comparación con el proceso manual.</p>

## **4.2 Procesamiento de datos**

### **4.2.1 Recolección de datos**

Los datos serán capturados mediante formularios digitales para garantizar la integridad de las respuestas. Se procedió a realizar la encuesta haciendo uso de la herramienta Microsoft Forms, la cual facilitó la distribución y el seguimiento de las respuestas correspondientes de manera eficiente y transparente. El tiempo promedio para llenar la encuesta fue de 3 minutos con 8 segundos. Teniendo en cuenta que el público objetivo responsable de evaluar la aplicación web integrada únicamente por 4 miembros de la directiva de la organización UNORICO SAMASHUNCHIK se define como un estudio con una muestra censal. En estudios donde la población es pequeña y cuenta con un conocimiento técnico o administrativo específico en este contexto, los reglamentos y procesos internos del cementerio de Otavalo, encuestar a todas las personas asegura una cobertura total del área de estudio. Por esta razón, el análisis de la información obtenida se tratará por medio del uso de estadística descriptiva. Esta estrategia justifica la omisión de evaluaciones con estadísticas inferenciales complejas, enfocándose en el cálculo de frecuencias y porcentajes, tal como mencionan Hernández et al.[64], lo cual permite mostrar de forma clara, objetiva y directa la evaluación de la directiva.

### **4.2.2 Verificación por estadística descriptiva**

La verificación de la aplicación web se enfocó en establecer el nivel de consenso y la calificación promedio concedidos por la directiva. De acuerdo con Hernández Sampieri et al. [64], La utilización de medidas de tendencia central, como la media y distribución de frecuencias, representa el método más adecuado para detallar y confirmar la tendencia de los datos al momento de evaluar a la población total o muestra censal. Dado que son los responsables directos de UNORICO SAMASHUNCHIK, su criterio garantizó que la aplicación cumpliera con los requerimientos verdaderos de la organización. Por lo tanto, por medio de la estadística descriptiva, se evaluó la media aritmética de las respuestas. Este método descriptivo fue el adecuado para la muestra censal. Permitiendo identificar de manera directa y objetiva el éxito de la aplicación para cada una de sus dimensiones.

El procesamiento de los datos se llevó a cabo utilizando el software estadístico IBM SPSS Statistics [65]. Por motivo de las cualidades de la muestra, el análisis omitió los índices de fiabilidad y se enfocó en un análisis descriptivo de variables individuales.

Para la interpretación de los datos obtenidos de la encuesta, se organizaron y calculó la media aritmética de cada dimensión del modelo de éxito de DeLone & McLean. Estas medias permitieron categorizar el rendimiento de la aplicación web en un rango establecido, apoyando en la toma de decisiones sobre la factibilidad y aprobación del software.

Para determinar de forma objetiva y estricta estos rangos de valoración, se procedió al cálculo de la amplitud de los intervalos para la escala de Likert que consiste en 5 puntos usada en la herramienta de evaluación. Para obtener la amplitud es restando el valor mínimo del máximo de la escala y dividiendo el resultado entre el número de categorías [66].

$$Amplitud = \frac{Valor\ Maximo - Valor\ Minimo}{N\ de\ categorias}$$

Aplicando esta fórmula, tenemos una amplitud de 0.80.

$$Amplitud = \frac{5 - 1}{5} = 0.80$$

Al conseguir un rango preciso de 0.80, se elaboraron las cinco categorías de manera proporcional y justa, empleando el cálculo estándar de amplitud de clase para distribuciones estadísticas [66]. Para normalizar la interpretación de los datos obtenidos por parte de la directiva, tomando como base a Matas [67], quien avala metodológicamente esta organización y adaptación de escalas en la evaluación de sistemas. El resultado de esta distribución de rangos de valoración se detalla en la Tabla 44.

*Tabla 44. Rangos de valoración según la Media.*

<b>Rango de la Media</b>	<b>Nivel de Valoración</b>
4.21 – 5.00	Excelente
3.41 – 4.20	Bueno
2.61 – 3.40	Aceptable
1.81 – 2.60	Regular
1.00 – 1.80	Deficiente

*Nota: Criterios adaptados para la interpretación de escalas tipo Likert en la evaluación de sistemas [67].*

A continuación, se presenta la tabla 45, se detallan los resultados de la encuesta ordenada por pregunta donde cada fila pertenece a un encuestado y su respuesta de acuerdo con la escala de Likert.

*Tabla 45. Resultados de la encuesta [Elaboración Propia]*

Encuestados	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
1	4	4	4	4	3	4	3	3	4	3	3	3	3	3	4	4	3	4	4	4	4	3
2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	4	4	3	4	4	4	4	4	4	4	4	5	5	4	4	4	4	5	5	4	4	4
4	5	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Después de tabular por medio de estadística descriptiva los datos obtenidos por parte de la directiva, se definieron las valoraciones globales del sistema. En la Tabla 46 se indican las medidas calculadas para cada dimensión del modelo de DeLone & McLean, demostrando el nivel de éxito y aceptación de la aplicación web para la gestión del cementerio por parte de los usuarios principales.

*Tabla 46 Resultados – Evaluación por dimensión [Elaboración Propia]*

<b>Dimensión</b>	<b>Ítems</b>	<b>Número de ítems (K)</b>	<b>Media de la dimensión</b>	<b>Interpretación</b>
<b>Calidad del sistema</b>	P1,P2,P3,P4	4	3.87	Bueno
<b>Calidad de la información</b>	P5,P6,P7,P8	4	3.81	Bueno
<b>Calidad del servicio</b>	P9,P10,P11	3	3.83	Bueno
<b>Uso</b>	P12,P13,P14	3	3.91	Bueno
<b>Satisfacción del Usuario</b>	P15,P16,P17	3	3.91	Bueno
<b>Impactos Netos</b>	P18,P19,P20, P21,P22	5	4.05	Bueno

### **4.3 Interpretación de datos**

En este apartado, se presentan y evalúan los resultados recopilados de la encuesta llevada a cabo para validar la aplicación web, diseñada para automatizar y gestionar la administración del cementerio indígena de Otavalo para la organización UNORICO SAMASHUNCHIK. Con base en el análisis de los datos obtenidos, se procura mejorar la experiencia del usuario en esta aplicación web con la automatización de procesos requeridos en el cementerio, al igual que mejorar la experiencia del usuario en estos procesos de gestión. Como se observa en la Tabla 46, los resultados adquiridos por parte de la directiva muestran una aceptación homogénea y favorable hacia la aplicación web, dado que las seis dimensiones evaluadas se colocaron dentro del rango de valoración “Bueno”. Justamente, las dimensiones Beneficios Netos, Uso y Satisfacción de Usuario resaltaron con las medias más altas de 4.05, 3.91 y 3.91 correspondientemente. Esto evidencia que la directiva nota una alta confiabilidad técnica y un beneficio directo para la administración del cementerio indígena.

En cambio, las otras dimensiones que también clasificaron en el rango “Bueno” como Calidad del sistema, calidad del servicio y calidad de la información registraron medias prudentes de 3.87, 3.83 y 3.81. Estos valores globales corroboran que la aplicación web cumple satisfactoriamente con las necesidades operativas de la organización, alcanzando el propósito de automatizar los procesos, a la vez que establecen un punto de partida para futuras actualizaciones que busquen mejorar la experiencia del usuario.

#### **4.3.1 Dimensiones del modelo DeLone & McLean**

En esta sección se analizaron los resultados obtenidos en cada una de las seis dimensiones de este modelo, lo cual se usó para examinar la calidad del sistema de información. Cada una de estas dimensiones representan una parte esencial que afecta la percepción general de los administrativos sobre el éxito de la aplicación. Se analizaron los hallazgos con relación a la satisfacción de los usuarios y la intención de uso con el objetivo de ofrecer una visión completa sobre la calidad de la aplicación web para el cementerio de Otavalo.

##### **4.3.1.1 *Calidad del sistema***

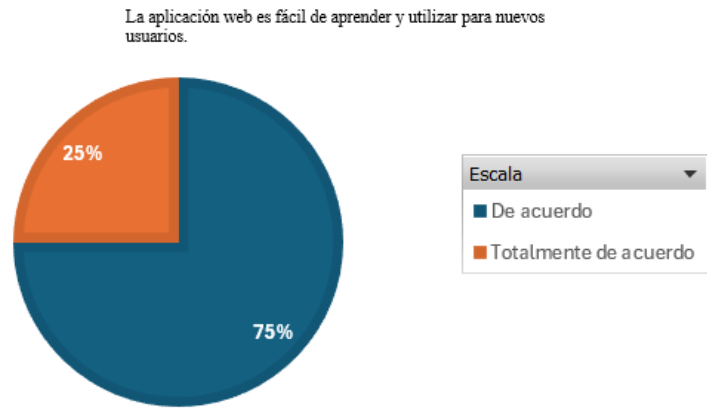


Figura 69 Calidad del Sistema. Pregunta 1. [Elaboración propia]

El análisis de la Calidad del Sistema refleja aceptación mayoritaria en el aspecto de facilidad de aprendizaje y uso de la aplicación web, obteniendo un 100% de valoraciones positivas clasificadas entre los rangos “De acuerdo” (75%) y “Totalmente de acuerdo” (25%). La total falta de enfoques neutrales o en desacuerdo verifica de forma decisiva la efectividad del diseño de la interfaz, comprobando la usabilidad de la aplicación como una herramienta totalmente intuitiva y de rápida adaptación para los usuarios del cementerio.

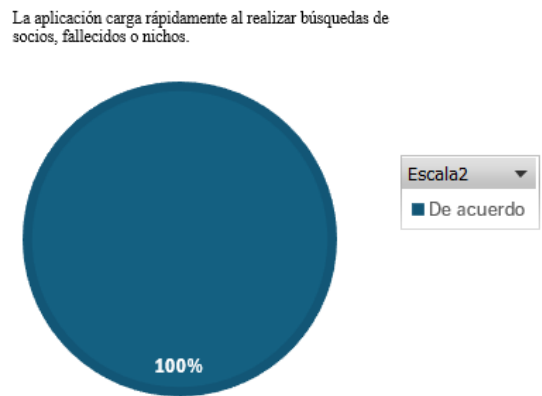


Figura 70 Calidad del Sistema. Pregunta 2. [Elaboración propia]

Con respecto al tiempo de carga y eficiencia de la aplicación, el 100% de los usuarios confirmó positivamente la rapidez del sistema al consultar socios, fallecidos o nichos, enfocándose la totalidad de las respuestas en el rango “De acuerdo”. La inexistencia total de respuestas neutrales y negativas valida que la arquitectura del

software y la eficiencia de la base de datos permiten que los usuarios de la organización logren efectuar sus consultas y obtener la información de forma rápida y eficiente.

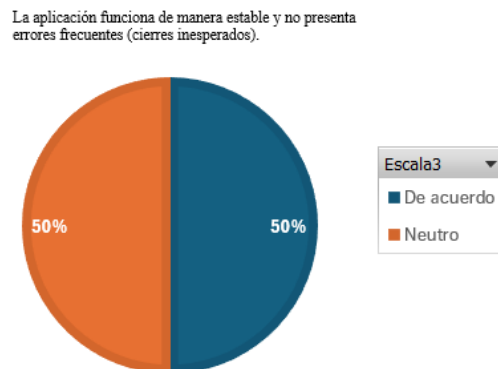


Figura 71 Sistema de Calidad. Pregunta 3. [Elaboración propia]

Al evaluar la estabilidad de la aplicación web, los resultados demuestran que el 50% de los usuarios confirmó el rendimiento del software, mostrando estar en “De acuerdo” en que la aplicación se ejecuta de forma estable y no revela errores recurrentes ni se detiene abruptamente. Mientras que el 50% restante tomó una posición “neutral” respecto a este indicador. La total falta de respuestas negativas verifica que la aplicación es factible y no sufre fallos graves que paren la gestión diaria de la organización. Aunque la mitad de los usuarios acepta la robustez de la aplicación, la influencia del sector neutral indica que la herramienta cumple con su propósito sin problemas, mientras el personal para por su proceso de aprendizaje.

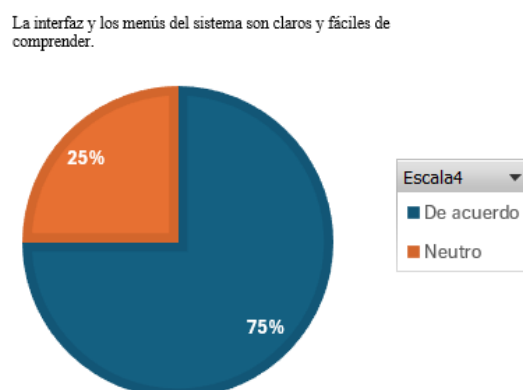


Figura 72 Sistema de Calidad. Pregunta 4. [Elaboración propia]

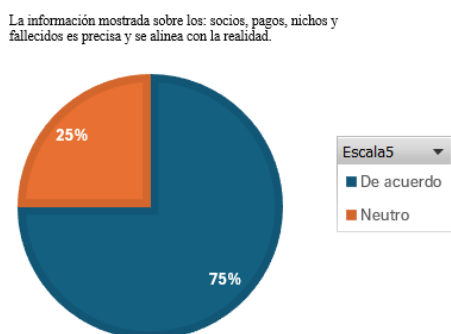
Al tratar el diseño funcional de la aplicación web, se mira una aprobación unánime por parte de los usuarios encuestados el 75% está “De acuerdo” que la interfaz es clara y sencilla de entender y el 25% tienen resultados en “Neutro” sin presentarse valores negativos o desaprobados. Esto señala que la gran mayoría de los usuarios aceptan que el diseño es intuitivo y fácil de entender.

*Tabla 47 Porcentaje de la Calidad de la Información. [Elaboración propia].*

Dimensión	Pregunta	Aprobación	Neutralidad	Desaprobación
Calidad del Sistema	P1	100%	0%	0%
	P2	100%	0%	0%
	P3	50%	50%	0%
	P4	75%	25%	0%
<b>Total</b>		81.25%	18.75%	0%

La Calidad del Sistema verifica de acuerdo con la tabla 47. Con estos resultados se verifica la factibilidad de la aplicación web, avalada por un índice de aprobación del 81.25%, un margen de neutralidad del 18.75% y con 0% de desaprobación. Estos resultados validan el éxito de la implementación técnica.

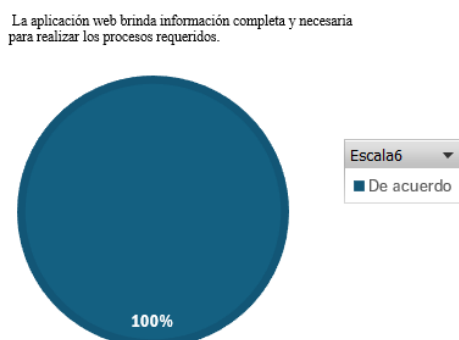
#### 4.3.1.2 Calidad de la Información



*Figura 73 Calidad de la Información. Pregunta 5. [Elaboración propia]*

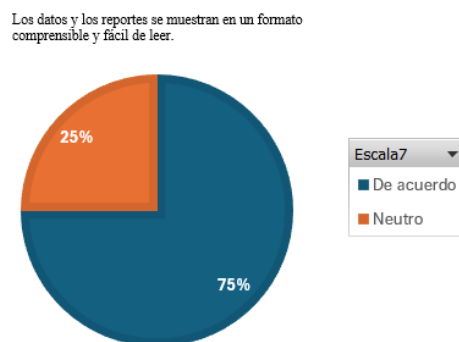
Al analizar la dimensión de la Calidad de la Información, los resultados muestran que el 75% de los usuarios están “De acuerdo” en que los datos obtenidos por la aplicación acerca de los nichos, fallecidos y socios son precisos y concuerdan con la realidad. Mientras un 25% son respuestas “Neutrales” y no existe presencia de respuestas en

desacuerdo, se observa que la aplicación web brinda información totalmente confiable, garantizando un control absoluto y preciso de los registros del cementerio.



*Figura 74 Calidad de la Información. Pregunta 6. [Elaboración propia]*

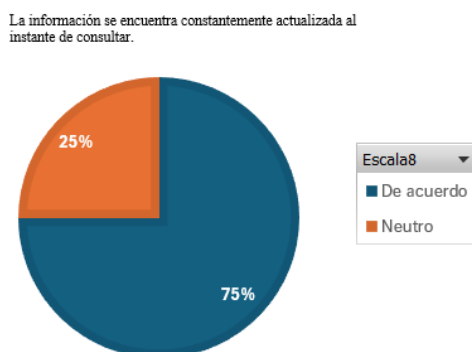
La evaluación con respecto a la totalidad y relevancia de los datos proporcionados por parte de la aplicación refleja una aprobación unánime del 100% enfocada totalmente en la valoración “De acuerdo”. Este resultado claro apoyado por la falta total de opiniones neutrales o en desacuerdo, corrobora que la aplicación ofrece de forma eficaz toda la información requerida para realizar los procesos necesarios. Esto simplifica la gestión operativa y administrativa con información completa.



*Figura 75 Calidad de la Información. Pregunta 7. [Elaboración propia]*

En términos de claridad y entendimiento de los reportes generados por la aplicación, los resultados demuestran una aceptación mayoritaria del 75% dentro del rango “De acuerdo”. Mientras con un 25% de indecisión entrando en el rango “Neutro” y no existen opiniones en desacuerdo, con esto se confirma que el diseño y la visualización de la información son fáciles de entender y usar. Esto garantiza que los

integrantes de la organización puedan manejar con facilidad la información de la aplicación web.



*Figura 76 Calidad de la Información. Pregunta 8. [Elaboración propia]*

Un factor clave para el funcionamiento de la organización se basa en la actualización de los datos. Los datos obtenidos por este indicador indican que el 75% de los encuestados se encuentran “De acuerdo” en que la aplicación muestra la información actualizada al momento de ser consultada. Con un 25% de opiniones neutrales y ninguna opinión en desacuerdo, se establece que la aplicación web asegura un flujo de datos en tiempo real. Esta característica es fundamental para la adecuada gestión del cementerio dado que permite a los responsables apoyarse en registros precisos y sincronizados.

*Tabla 48 Porcentaje - Calidad de la Información [Elaboración propia].*

<b>Dimensión</b>	<b>Pregunta</b>	<b>Aprobación</b>	<b>Neutralidad</b>	<b>Desaprobación</b>
<b>Calidad de la Información</b>	P5	75%	25%	0%
	P6	100%	0%	0%
	P7	75%	25%	0%
	P8	75%	25%	0%
<b>Total</b>		81.25%	18.75%	0%

La Calidad de la Información indica una aceptación por parte de los usuarios con un índice de aprobación del 81.25%, un 18.75% de resultados neutrales y un porcentaje de 0% en desaprobación como se refleja en la tabla 48. Con estos resultados se verifica que la aplicación proporciona información oportuna y segura para la organización.

### 4.3.1.3 Calidad de Servicio

La aplicación web muestra mensajes de ayuda o validación claros cuando ocurre o comete un error.

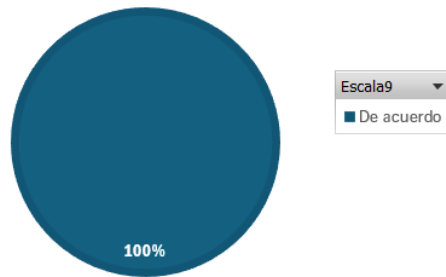


Figura 77 Calidad de Servicio. Pregunta 9. [Elaboración propia]

Al analizar la Calidad de Servicio desde el punto de vista de interacción y administración de errores, los resultados demuestran que mayoritariamente el 100% se encuentran "De acuerdo" en que la aplicación cumple notificando mensajes de validación de manera clara y adecuada. La presencia de valores nulos con respecto a opiniones neutras y desfavorables demuestra que la aplicación dispone de mecanismos de respuesta eficientes para el usuario. Esto garantiza que los miembros de la organización tengan una retroalimentación correcta y necesaria para mejorar en sus acciones.

Cuenta con la ayuda o la orientación necesaria para solucionar dudas sobre el uso de la aplicación web.

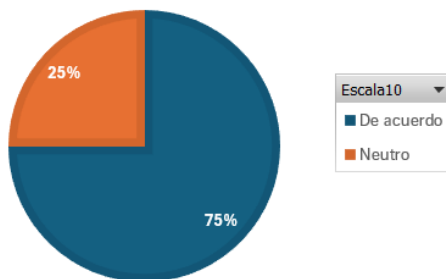


Figura 78 Calidad del Servicio. Pregunta 10. [Elaboración propia]

Respecto a brindar ayuda y orientación acerca de la aplicación web, los resultados muestran que el 75% del personal están "De acuerdo" en que cuentan con la guía requerida. Y el 25% se muestra en "Neutro", demostrando una opinión indecisa acerca de la orientación mencionada. La falta de respuestas desfavorables indica que la aplicación cuenta con un soporte efectivo, lo que asegura una retroalimentación apropiada para perfeccionar la experiencia del usuario.

La capacitación o indicaciones recibidas son suficientes para manejar la aplicación correctamente.

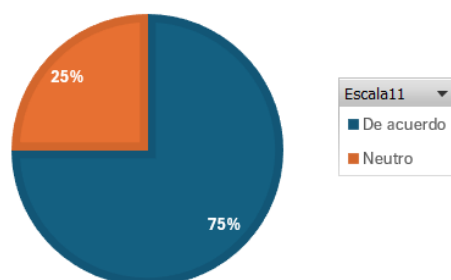


Figura 79 Calidad del Servicio. Pregunta 11. [Elaboración propia]

Al analizar la capacitación recibida para usar adecuadamente la aplicación, los resultados demostraron que el 75% de los usuarios están "De acuerdo" con que las instrucciones son adecuadas. Mientras que el 25% se refleja en "Neutro", demostrando una posición neutral acerca de la eficacia de la capacitación. La falta total de respuestas negativas indica que la aplicación cuenta con personal de entrenamiento eficiente, lo que asegura una retroalimentación correcta para perfeccionar la experiencia del usuario y garantizar el uso apropiado de la aplicación.

Tabla 49 Porcentaje - Calidad de Servicio. [Elaboración propia]

Dimensión	Pregunta	Aprobación	Neutralidad	Desaprobación
Calidad de Servicio	P9	100%	0%	0%
	P10	75%	25%	0%
	P11	75%	25%	0%
<b>Total</b>		<b>83.33%</b>	<b>16.66%</b>	<b>0%</b>

La Calidad de Servicio demuestran aceptación favorable, teniendo un índice de aprobación del 83.33%. Igualmente se muestra un 16.66% de neutralidad y el nivel de desaprobación es nulo, teniendo un 0%, esto se refleja en la tabla 49. Con estos resultados se verifica que la aplicación brinda un nivel de servicio eficiente para la gestión del cementerio.

#### 4.3.1.4 Uso / Intención de Uso

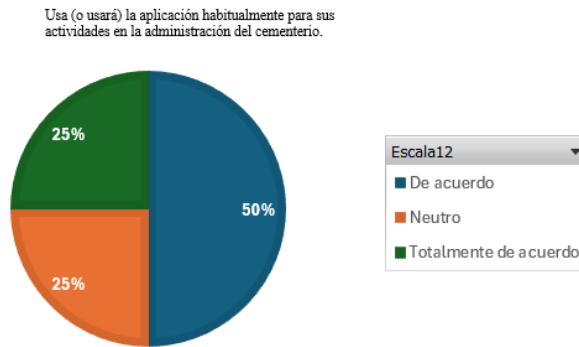


Figura 80 Uso-Intención de Uso. Pregunta 12. [Elaboración propia]

En cuanto al Uso - intención de uso sobre la aplicación para la gestión del cementerio, los resultados muestran que un 75% de los usuarios confirman que usarán la aplicación habitualmente clasificando entre las categorías "De acuerdo" y "Totalmente de acuerdo" con valoración de 50% y 25%, y el otro 25% se indica en "Neutro" reflejando una opinión indecisa sobre el uso frecuente de la aplicación. Esto demuestra la aceptación de la aplicación como herramienta funcional para las tareas del cementerio.

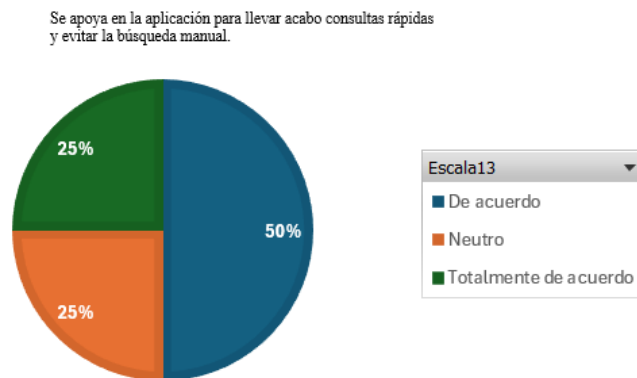


Figura 81 Uso-Intención de Uso. Pregunta 13. [Elaboración propia]

Al analizar las consultas rápidas que se realizaron en la aplicación web para evitar las consultas manuales, los resultados demuestran un 75% de favorabilidad por parte de los usuarios, el 50% es que se encuentran "De acuerdo" y el otro 25% "Totalmente de acuerdo" de que las consultas son eficientes. Mientras que el restante 25% se mantiene en "Neutro" y no hay resultados negativos indicando una fuerte aceptación de la

aplicación para agilizar las consultas y mejora la productividad en la gestión del cementerio.

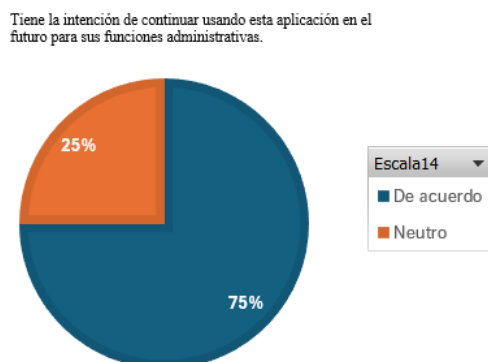


Figura 82 Uso-Intención de Uso. Pregunta 14. [Elaboración propia]

Con respecto a la intención de continuar usando la aplicación web en el futuro para las actividades administrativas se muestra que el 75% del personal está “De acuerdo” con continuar usándola. Y el 25% indica resultados en “Neutro”, no hubo rechazo ni valoraciones negativas, lo que define que la gran mayoría de los usuarios nota que la aplicación es una herramienta útil para mejorar las tareas administrativas del cementerio en el futuro.

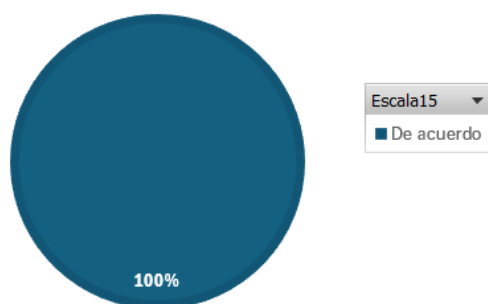
Tabla 50 Porcentaje - Uso - Intención de Uso. [Elaboración propia]

Dimensión	Pregunta	Aprobación	Neutralidad	Desaprobación
Uso – Intención de Uso	P1	75%	25%	0%
	P2	75%	25%	0%
	P3	75%	25%	0%
<b>Total</b>		75%	25%	0%

El Uso - Intención de Uso demuestran una calificación del 75% de aceptación positiva, tomando en cuenta que el 25% de las respuestas son posturas neutrales y se conserva la tasa de desaprobación nula del 0%. Esto evidencia una actitud positiva hacia el uso de la aplicación, lo cual permite aprobar esta dimensión con éxito. Como se presenta en la tabla 50.

#### 4.3.1.5 Satisfacción del Usuario

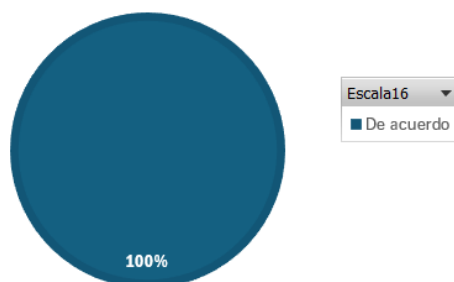
Esta satisfecho con el rendimiento de la aplicación en comparación con las operaciones manuales anteriores.



*Figura 83 Satisfacción del Usuario. Pregunta 15. [Elaboración propia]*

Al analizar la Satisfacción del usuario con respecto al rendimiento de la aplicación frente a las tareas manuales, los resultados demuestran que el 100% del personal administrativo está “De acuerdo” en que la aplicación web aumenta el rendimiento. Esto indica una aprobación completa y sensación de optimización al reemplazar procesos manuales por la herramienta digital.

La aplicación satisface con todas sus expectativas para la gestión del cementerio.



*Figura 84 Satisfacción del Usuario. Pregunta 16. [Elaboración propia]*

Al evaluar si la aplicación cumple con todas las necesidades para la gestión del cementerio, se muestra que el 100% de los usuarios están “De acuerdo” en que la aplicación cumple con sus expectativas. Esto refleja que la herramienta es considerada como un apoyo clave para la administración del cementerio.

Se siente cómodo y seguro interactuando con esta aplicación web.

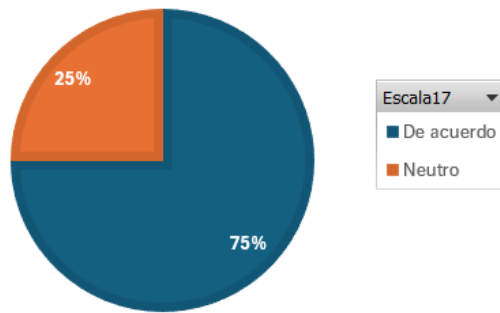


Figura 85 Satisfacción del Usuario. Pregunta 17. [Elaboración propia]

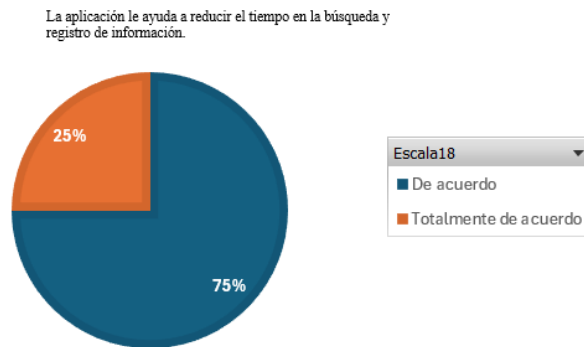
Al analizar si la aplicación permite sentirse cómodo y seguro cuando se interactúa con ella, el 75% del personal administrativo mencionó que está “De acuerdo”, mientras que el 25% menciona que están en “Neutro”. Esto demuestra una acogida positiva, pero hay una parte que se mantiene con dudas sobre el uso de la aplicación.

Tabla 51 Porcentaje - Satisfacción de Usuario. [Elaboración propia]

Dimensión	Pregunta	Aprobación	Neutralidad	Desaprobación
Satisfacción del Usuario	P1	100%	0%	0%
	P2	100%	0%	0%
	P3	75%	25%	0%
<b>Total</b>		91.66%	8.33%	0%

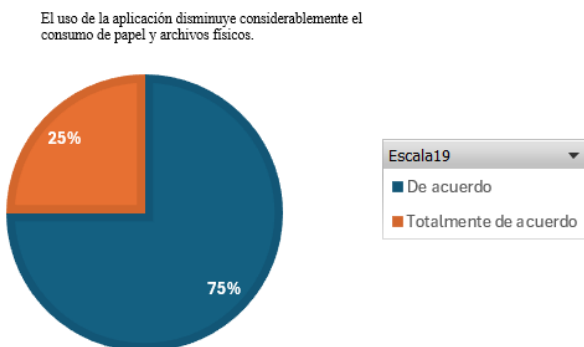
La Satisfacción del Usuario; los resultados demuestran una acogida positiva de la aplicación. En general esta dimensión logró una calificación global de 91.66% como se evidencia en la tabla 51. Con estos resultados se verifica que es satisfactorio y cumple con las expectativas de los usuarios.

#### 4.3.1.6 Beneficios Netos



*Figura 86 Beneficios Netos. Pregunta 18. [Elaboración propia]*

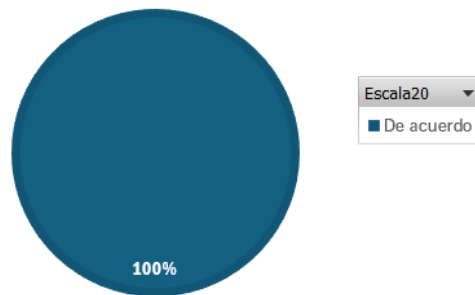
Al analizar los beneficios netos con respecto a la reducción de tiempo en búsqueda y registro de datos, el 75% de los usuarios indican que están “De acuerdo” y el otro 25% demuestra que están “Totalmente de acuerdo”, no se tienen datos negativos o en desacuerdo. Esto demuestra que la aplicación es vista como eficiente ya que disminuye el tiempo para los procesos de búsqueda y registro de información.



*Figura 87 Beneficios Netos. Pregunta 19. [Elaboración propia]*

Al examinar si el uso de la aplicación disminuye notablemente el uso del papel y tener archivos físicos, se muestra que el 75% del personal está “De acuerdo” y el resto del 25% está “Totalmente de acuerdo” sin tener resultados neutrales y negativos, nos demuestra que la directiva del cementerio acepta completamente que la aplicación será de gran ayuda para la reducción del uso continuo del papel.

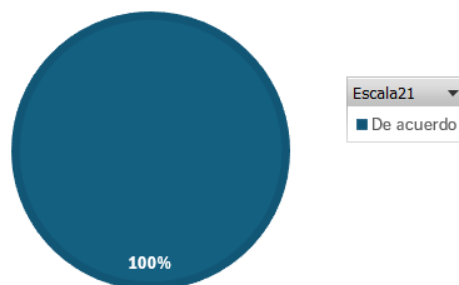
La aplicación mejora la transparencia, la gestión de pagos, nichos, fallecidos y socios.



*Figura 88 Beneficios Netos. Pregunta 20. [Elaboración propia]*

Al analizar si la aplicación mejora la transparencia y la administración de los nichos, pagos, socios y fallecidos, los resultados indican que el 100% de los usuarios están “De acuerdo”, no se observan resultados neutros y en desacuerdo. Esto demuestra que la aplicación web es tomada en cuenta como una solución muy eficiente para mejorar las tareas administrativas y crear confianza en la gestión de datos.

La aplicación ayuda a prevenir equivocaciones o fallos como la duplicidad de registros o pérdida de información.



*Figura 89 Beneficios Netos. Pregunta 21. [Elaboración propia]*

Al analizar si la aplicación ayuda a evitar errores o fallos como la duplicidad de datos o la pérdida de datos, se observa que el 100% de los usuarios están “De acuerdo”, no se tiene datos neutros o negativos. Esto evidencia que la aplicación tiene un alto nivel de seguridad y confiabilidad en la protección de datos, reduciendo de manera efectiva los errores humanos comunes.

El código identificador facilita el proceso de búsqueda en comparación con el proceso manual.

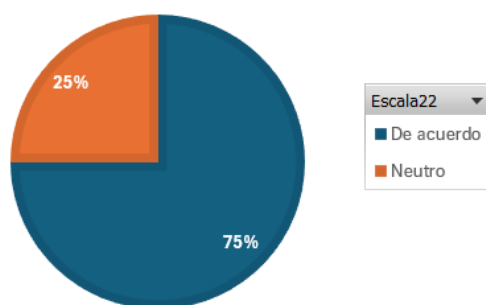


Figura 90 Beneficios Netos. Pregunta 22. [Elaboración propia]

Al evaluar si el código identificador agiliza el proceso de consulta comparando con el trabajo manual, se revela que el 75% de los encuestados están “De acuerdo”, y el 25% se conserva en “Neutro” sin datos negativos. Esto evidencia que la mayoría considera esta característica como un beneficio considerable en la productividad, simplificando y mejorando significativamente el tiempo empleado en buscar información a diferencia de los métodos clásicos.

Tabla 52 Porcentaje - Beneficios Netos. [Elaboración propia]

Dimensión	Pregunta	Aprobación	Neutralidad	Desaprobación
<b>Beneficios Netos</b>	P18	100%	0%	0%
	P19	100%	0%	0%
	P20	100%	0%	0%
	P21	100%	0%	0%
	P22	75%	25%	0%
<b>Total</b>		95%	5%	0%

En la tabla 52 se representa los resultados obtenidos de la dimensión Beneficios Netos del modelo de DeLone y McLean, donde los ítems P18-P22 lograron una aceptación promedio del 95%, indecisión del 5% y desaprobación del 0%, lo que señala que los usuarios reconocen claramente que la aplicación web brinda beneficios fundamentales como simplificar la gestión de pagos, nichos, socios y fallecidos, mejorar la transparencia, también impide la duplicidad de datos y pérdida, pero se ve un poco de duda en la parte relacionado con el uso del código identificador para los procesos de búsqueda, lo que da

a entender la importancia de mejorar la usabilidad de esta funcionalidad u ofrecer más capacitación interactiva para mejorar la percepción.

#### 4.4 Análisis de resultados

El análisis final de cada dimensión permite compilar los resultados de datos recopilados en la evaluación de la aplicación para la gestión del cementerio indígena de Otavalo, clasificando las opiniones de los usuarios por la clasificación definida en el modelo de éxito de DeLone y McLean.

Comenzando por los porcentajes de aprobación, neutralidad y desaprobación, se obtienen conclusiones que dirigen planes de acción para mejorar la usabilidad y fortalecer el valor de la aplicación.

*Tabla 53 Análisis por dimensión del modelo DeLone y McLean. [Elaboración propia]*

<b>Dimensión</b>	<b>Aprobación</b>	<b>Neutralidad</b>	<b>Desaprobación</b>
<b>Calidad del Sistema</b>	81.25%	18.75%	0%
<b>Calidad de la Información</b>	81.25%	18.75%	0%
<b>Calidad del Servicio</b>	83.33%	16.66%	0%
<b>Uso – Intención de Uso</b>	75%	25%	0%
<b>Satisfacción de Usuario</b>	91.66%	8.33%	0%
<b>Beneficios Netos</b>	95%	5%	0%

Este análisis final, ilustrado en la tabla 53, por cada una de las dimensiones prueba una aceptación positiva de la aplicación para la gestión del cementerio en la ciudad de Otavalo: las dimensiones de Beneficios Netos, Satisfacción de Usuario consiguen más del 90% de aprobación cada una, no se presentan niveles de desaprobación. Estos resultados resumen que la aplicación es vista como intuitiva y segura, que la información

que proporciona es útil y adecuada, y que la influencia organizacional rompe límites lo cual optimiza la experiencia y facilita su búsqueda.

## **Conclusiones y recomendaciones**

### **Conclusiones**

La revisión bibliográfica permitió definir un marco conceptual firme necesario para el desarrollo de aplicaciones web, dando énfasis a la digitalización de procesos administrativos dirigidos a asociaciones comunitarias. El estudio acerca de tecnologías de desarrollo como Laravel, PostgreSQL Supabase guio la creación de una herramienta adecuada a los requerimientos administrativos del cementerio indígena de Otavalo. La fundamentación teórica permitió definir un cimiento técnico sólido para el desarrollo de la aplicación web, centrada en solucionar los requerimientos de la gestión administrativa de asociaciones comunitarias. La revisión bibliográfica respaldó la elección de tecnologías innovadoras y eficaces, como el Framework Laravel y la administración de la base de datos PostgreSQL almacenada en Supabase. Igualmente, la integración del marco de trabajo ágil SCRUM y del modelo de evaluación de DeLone y McLean estableció la estructura metodológica esencial para orientar el desarrollo de forma iterativa y medir con precisión el efecto de la aplicación en la organización UNORICO SAMASHUNCHIK.

La implementación de la aplicación web se ejecutó de manera exitosa, consiguiendo digitalizar y unificar los procesos del cementerio. El software facilita a la directiva gestionar de forma óptima el registro de fallecidos, socios, asignaciones, control de fallecidos, asignación de nichos y el monitoreo de pagos. Desde la perspectiva técnica, la aplicación se desarrolló con el patrón de diseño MVC (Modelo-Vista-Controlador), empezando su codificación en un ambiente de desarrollo local con XAMPP y finalizando con la exitosa migración de la base de datos a la nube por medio de Supabase en la fase final del proyecto. Esta estructura garantiza la confiabilidad, seguridad y accesibilidad de los datos constantemente.

La evaluación de la calidad del sistema, hecha a través de una muestra censal a la directiva del cementerio aplicando el modelo de DeLone y McLean, demostró ser altamente factible y la adopción de la herramienta consiguió un 0% de desaprobación en

todas las dimensiones. Se resaltaron niveles excepcionales de aceptación en Beneficios Netos con un 95% y la Satisfacción del Usuario con un 91.66%, lo que valida que la aplicación brinda claridad y evita la pérdida de información. Aunque se observaron porcentajes neutros en la Calidad del Sistema con 18.75% y en el Uso e Intención de Uso con 25%, estos datos no significan deficiencias en el código, sino que muestran el comportamiento habitual de acoplamiento de los usuarios al transformar el modelo manual en digital.

## **Recomendaciones**

Se sugiere a la organización UNORICO SAMASHUNCHIK actualizar regularmente las versiones de las tecnologías empleadas, precisamente el Framework Laravel y los servicios de almacenamiento de base de datos en Supabase. Planificar revisiones frecuentes permitirá aprovechar mejoras de eficiencia y actualizaciones de seguridad. Igualmente, para la creación de futuros módulos, se recomienda configurar entornos de testing de forma directa en la nube desde las fases iniciales, minimizando la dependencia de servidores locales para optimizar el proceso de despliegue.

Para minimizar la incertidumbre detectada a lo largo de la evaluación, se recomienda reforzar la capacitación tecnológica del personal administrativo del cementerio. Se sugiere agregar ayudas visuales directamente en el panel de administración, por ejemplo, notificaciones, alertas o un manual de usuario digital incorporado, que orienten punto por punto a los usuarios en las tareas que despertaron mayor interés, como el uso de códigos identificadores para las búsquedas de los nichos con sus respectivos fallecidos y socios.

Aprovechando la potente estructura de la base de datos relacional elaborada, se propone mejorar el análisis de la aplicación web en la siguiente etapa del proyecto. Se sugiere la creación de paneles de control que creen reportes automáticos y predictivos acerca del nivel de uso del espacio de los nichos e informen sobre los retrasos en los pagos de los socios. Estas herramientas ofrecerán a la directiva métricas importantes que ayudarán a mejorar la toma de decisiones administrativas.

## Bibliografía

- [1] J. Alexander *et al.*, “SISTEMA DE CATASTRO PARA LA ALCALDIA DE CUSCATANCINGO,” 2019.
- [2] J. D. ALARCÓN JIMENES, “SISTEMA DE INFORMACIÓN CATASTRAL PARA LA GESTIÓN DEL CEMENTERIO MUNICIPAL DE LA CIUDAD DE PUYO MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.,” Puyo, 2019.
- [3] A. De Uso and Y. Publicación, “DESARROLLO DE UN SISTEMA WEB PARA FORTALECER LA GESTIÓN Y EL CONTROL DEL PROCESO PRODUCTIVO DE LA EMPRESA PÚBLICA MUNICIPAL DE FAENAMIENTO Y PRODUCTOS CÁRNICOS DE IBARRA,” 2025.
- [4] M. de los Á. Ruano Lara, “SISTEMA DE REGISTRO DE BIENES INMUEBLES ANCESTRALES DE LA PARROQUIA SAN ANTONIO DE IBARRA UTILIZANDO EL FRAMEWORK LARAVEL Y UN VISOR GEOGRÁFICO,” Ibarra, 2020.
- [5] “Laravel - the PHP framework for web artisans,” Laravel.com. Accessed: Apr. 28, 2025. [Online]. Available: <https://laravel.com/>
- [6] “PostgreSQL,” PostgreSQL. Accessed: Apr. 28, 2025. [Online]. Available: <https://www.postgresql.org/>
- [7] D. T. T. Q. H. V. and L. A. T. N. Thanh Son Huynh, “Design and Implementation of Web Application Based on MVC Laravel Architecture,” Aug. 2022, doi: 10.24018/ejece.2021.6.4.448.
- [8] A. Santiago and L. Borja, “Desarrollo de una aplicación web para mejorar el control y gestión de los procesos de producción en la empresa textil Acuatex.,” Ibarra, 2024.
- [9] “Mendeley Reference Manager,” <https://www.mendeley.com/reference-management/reference-manager>.
- [10] S. M. BoK, “Modelo original de Scrum para desarrollo de software,” Scrum Manager BoK. Accessed: Apr. 29, 2025. [Online]. Available: [https://www.scrummanager.com/bok/index.php/Modelo\\_original\\_de\\_Scrum\\_para\\_desarrollo\\_de\\_software](https://www.scrummanager.com/bok/index.php/Modelo_original_de_Scrum_para_desarrollo_de_software)
- [11] M. Moran, “Desarrollo Sostenible,” Infraestructura.
- [12] “Plan-de-Creación-de-Oportunidades-2021-2025-Aprobado,” 2021.
- [13] C. Antonio Martín Moreno *et al.*, “POLÍTICA PÚBLICA PARA LA TRANSFORMACIÓN DIGITAL DEL ECUADOR 2025-2030,” 2025.
- [14] DENNYS ANDRÉS OCAÑA LEMA, “IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO INFORMÁTICO Y GENERACIÓN DE

- CÓDIGO QR, PARA LA FACULTAD DE MECÁNICA EN LA ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, DESARROLLADO EN LARAVEL Y GESTOR DE BASE DE DATOS MYSQL EN EL PERÍODO 2020.,” 2021, Accessed: Apr. 30, 2025. [Online]. Available: <https://sangabrielriobamba.edu.ec/tesis/sistemas/tesis27.pdf>
- [15] Shirley Mónica Adriano Adriano, “Sistema web de gestión de información para el cementerio del GAD de Guano utilizando Framework CodeIgniter.,” Riobamba, Jun. 2022. Accessed: Apr. 25, 2025. [Online]. Available: <http://dspace.unach.edu.ec/handle/51000/9306>
- [16] KEVIN ANDRÉS ARROYAVE MURILLO, “APLICACIÓN WEB PARA LA GESTIÓN DE LA INFORMACIÓN DE LOS PROCESOS ADMINISTRATIVOS EN LOS CEMENTERIOS MUNICIPALES - CEMTECH. CASO DE ESTUDIO: CEMENTERIO MUNICIPAL DE GRANADA, ANTIOQUIA,” 2024.
- [17] F. De and C. Administrativas, “MODELO DE GESTIÓN ADMINISTRATIVA SUSTENTADO EN LAS COMPETENCIAS Y ATRIBUCIONES DE LOS GOBIERNOS AUTÓNOMOS DESCENTRALIZADOS MUNICIPALES PARA EL CEMENTERIO GENERAL DEL CANTÓN SANTA ELENA, AÑO 2013,” 2013.
- [18] B. Chibuye and J. Phiri, “An Assessment for Record Keeping and Cemetery Management: A Case of Lusaka City Council, Zambia Based on Technology Acceptance Model (TAM),” *Open Journal of Business and Management*, vol. 10, no. 02, pp. 577–590, 2022, doi: 10.4236/ojbm.2022.102032.
- [19] Z. A. Horn, “Cemeteries & the Control of Bodies.”
- [20] SALGA, “GOOD PRACTICES IN CEMETERIES MANAGEMENT Building Local Government Capacity in Cemeteries Management,” 2016.
- [21] A. Cajas Klever Xavier Cisneros Evelyn Karen, “IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN PARA EL CEMENTERIO CENTRAL DEL GAD MUNICIPAL DEL CANTÓN LA MANÁ,” Sep. 2016.
- [22] G. " De, L. C. De Tulcán, J. Andrés, and G. Bustos, “DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG) PARA EL CEMENTERIO ‘JOSÉ MARÍA AZAÉL FRANCO GUERRERO’ DE LA CIUDAD DE TULCÁN.”
- [23] E. E. Paredes Campoverde, “Solución de software para gestionar y geolocalizar datos de catastro - Estudio de caso: Cementerio Municipal de la provincia de Sucumbíos, cantón Cuyabeno,” Loja, Jan. 2024.
- [24] M. L. Schmidt, F. Yuan, and W. Jang, “Cemetery Mapping and Digital Data Analysis: A Case Study in Minnesota, USA,” *Journal of Geography and Geology*, vol. 12, no. 2, p. 40, Aug. 2020, doi: 10.5539/jgg.v12n2p40.
- [25] N. Demir and N. Yogeswaran, “Semi-automated cemetery mapping using smartphones,” in *International Archives of the Photogrammetry, Remote Sensing*

*and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Nov. 2018, pp. 59–62. doi: 10.5194/isprs-archives-XLII-5-59-2018.

- [26] A. Niarman, Iswandi, and A. K. Candri, “Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing,” *International Journal Software Engineering and Computer Science (IJSECS)*, vol. 3, no. 3, pp. 424–436, Dec. 2023, doi: 10.35870/ijsecs.v3i3.1850.
- [27] P. De Titulación Previo, A. La Obtención, and D. Título, “IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE SERVICIOS TÉCNICOS A DOMICILIO Y CONTROL DE INVENTARIO STOCK DE INSUMOS Y MATERIALES, UTILIZANDO EL FRAMEWORK PHP LARAVEL Y GESTOR DE BASE DE DATOS MYSQL, EN EL CENTRO COMERCIAL MACHADO, UBICADO EN LA CIUDAD DE RIOBAMBA EN EL PERIODO 2021.”
- [28] Loek van der Linde, “CHARACTERIZING JUNIOR CS STUDENT KNOWLEDGE OF MODEL-VIEW-CONTROLLER WHAT DO STUDENTS UNDERSTAND FROM MVC?,” Apr. 2024. [Online]. Available: <https://research.ou.nl/>
- [29] Quang Hung Nguyen, “BUILDING A WEB APPLICATION WITH LARAVEL 5,” 2015.
- [30] S. D. C. M. de la L. V. R. T. de J. J. B. Raúl Avalos Rojas, “Revisión de Framework Laravel y su aplicación en sistema web de mesa de ayuda,” vol. 15, Mar. 2008.
- [31] “MVC design pattern,” GeeksforGeeks.
- [32] “Introduction - Laravel 4.2 - The PHP Framework For Web Artisans,” <https://laravel.com/docs/4.2/introduction?>
- [33] “Installation - Laravel 12.x - The PHP Framework For Web Artisans,” <https://laravel.com/docs/12.x/installation>.
- [34] The PostgreSQL Global Development Group, “Acerca de PostgreSQL,” <https://www.postgresql.org/about/>.
- [35] The PostgreSQL Global Development Group, “2. A Brief History of PostgreSQL,” <https://www.postgresql.org/docs/current/history.html>.
- [36] R. Rea Wellington Israel, “DESARROLLO DE UNA GUÍA METODOLÓGICA PARA LA GESTIÓN DE PROYECTOS DE SOFTWARE CON LA HERRAMIENTA GITHUB Y EL MARCO DE TRABAJO SCRUM.”
- [37] A. De Uso, Y. Publicación, and A. F. De, “DESARROLLO DE UNA GUÍA METODOLÓGICA PARA MEJORAR LA DOCUMENTACIÓN DEL CAPÍTULO TÉCNICO DE LOS TRABAJOS DE GRADO DE LA CARRERA DE SOFTWARE QUE UTILICEN LA METODOLOGÍA SCRUM.”

- [38] W. H. DeLone and E. R. McLean, “The DeLone and McLean model of information systems success: A ten-year update,” in *Journal of Management Information Systems*, M.E. Sharpe Inc., 2003, pp. 9–30. doi: 10.1080/07421222.2003.11045748.
- [39] A. Jeyaraj, “DeLone & McLean models of information system success: Critical meta-review and research directions,” *Int. J. Inf. Manage.*, vol. 54, Oct. 2020, doi: 10.1016/j.ijinfomgt.2020.102139.
- [40] W. H. DeLone and E. R. McLean, “Information Systems Success Measurement,” *Foundations and Trends® in Information Systems*, vol. 2, no. 1, pp. 1–116, 2016, doi: 10.1561/29000000005.
- [41] T. Widyaningrum, Q. Sholihah, and B. S. Haryono, “The Delone and McLean Information System Success Model: Investigating User Satisfaction in Learning Management System,” *Journal of Education Technology*, vol. 8, no. 1, pp. 86–94, Apr. 2024, doi: 10.23887/jet.v8i1.71080.
- [42] Y. F. Viriando and Sfenrianto, “Using Delone & Mclean information system success model to evaluate the success of online platform,” *Journal of System and Management Sciences*, vol. 11, no. 2, pp. 182–198, 2021, doi: 10.33168/JSMS.2021.0212.
- [43] M. Begoña, “Validity of DeLone and McLean’s Model of Information Systems success at the web site level of analysis,” 2010. doi: 10.31390/gradschool\_dissertations.3531.
- [44] “Delone and McLean IS success model,” Information Systems Theories.
- [45] A. I. Ojo, “Validation of the delone and mclean information systems success model,” *Healthc. Inform. Res.*, vol. 23, no. 1, pp. 60–66, Jan. 2017, doi: 10.4258/hir.2017.23.1.60.
- [46] M. H. Rahman *et al.*, “Design, development and implementation of a digital burial record-keeping and management system in Bangladesh,” *J. Glob. Health*, vol. 15, p. 04086, Feb. 2025, doi: 10.7189/jogh.15.04086.
- [47] H. Apriyanto and C. D. Anggraeni, “Integrated Identity Tag System for Biodiversity Collections in Indonesian Botanical Garden Using QR Code,” *SISTEMASI*, vol. 13, p. 898, May 2024, doi: 10.32520/stmsi.v13i3.3225.
- [48] M. H. Rahman, M. Naderuzzaman, M. A. Kashem, B. M. Salahuddin, and M. Z. Mahmud, “Comparative Study: Performance of MVC Frameworks on RDBMS,” *International Journal of Information Technology and Computer Science*, vol. 16, pp. 26–34, Feb. 2024, doi: <https://doi.org/10.5815/ijitcs.2024.01.03>.
- [49] B. Campos Montero, D. Vigo Rodríguez, C. Rodríguez Sandoval, J. Santos Fernández, and R. Sánchez Ticona, “Sistema web eficiente para la gestión administrativa en una empresa de transportes privada,” *Revista peruana de computación y sistemas*, vol. 6, pp. 39–46, Jun. 2024, doi: 10.15381/rpcs.v6i1.27418.

- [50] Mangapul Siahaan and R. W. Wijaya, "Performance Comparison Between Laravel and ExpressJs Framework Using Apache JMeter," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 7, pp. 545–554, Jan. 2024, doi: 10.31289/jite.v7i2.10571.
- [51] P. S. Darmalaksana, A. C. Nugroho, and FX. H. Prasetya, "Website-Based Vintage Clothing Sales Information System Using the Laravel Framework," *Journal of Business and Technology*, vol. 5, pp. 8–18, Mar. 2025, doi: <https://doi.org/10.24167/jbt.v5i1.11388>.
- [52] Nailul Hikmi and Fitni Hidayati, "System Quality and the Impact of the Use of the Flo Application on Free Association Within the Community in Padang City Using the Delone and Mclean Method," *Professional Evidence-based Research and Advances in Wellness and Treatment*, vol. 2, pp. 12–22, Apr. 2025, doi: <https://doi.org/10.69855/perawat.v2i1.113>.
- [53] E. Sofiani, E. D. S. Mulyani, and T. Mufizar, "ANALISIS KEBERHASILAN PENGGUNAAN EDUCATION MANAGEMENT INFORMATION SYSTEM (EMIS) 4.0 MENGGUNAKAN METODE DELONE AND MCLEAN," *JRIS : Jurnal Rekayasa Informasi Swadharma*, vol. 5, pp. 108–118, Jan. 2025, doi: <https://doi.org/10.56486/jris.vol5no1.713>.
- [54] A. De Uso, Y. Publicación, and A. F. De, "Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de Software que utilicen la metodología SCRUM," Mar. 2024, Accessed: Oct. 03, 2024. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/15812>
- [55] K. Schwaber and J. Sutherland, "The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game," 2020.
- [56] "¿Qué es MVC? - Laravel - El framework PHP para artesanos web." Accessed: Feb. 15, 2026. [Online]. Available: <https://laravel.com/learn/getting-started-with-laravel/what-is-mvc>
- [57] "Blade Templates - Laravel 12.x - The PHP Framework For Web Artisans." Accessed: Feb. 15, 2026. [Online]. Available: <https://laravel.com/docs/12.x/blade>
- [58] "Middleware - Laravel 12.x - The PHP Framework For Web Artisans." Accessed: Feb. 15, 2026. [Online]. Available: <https://laravel.com/docs/12.x/middleware#main-content>
- [59] "Eloquent: Getting Started - Laravel 12.x - The PHP Framework For Web Artisans." Accessed: Feb. 15, 2026. [Online]. Available: <https://laravel.com/docs/12.x/eloquent>
- [60] "PostgreSQL: Documentation: 18: 28.1. Reliability." Accessed: Feb. 15, 2026. [Online]. Available: <https://www.postgresql.org/docs/current/wal-reliability.html>
- [61] C. Las Almas Y Bolívar and U. Samashunchik, "UNORICO SAMASHUNCHIK."

- [62] M. N. Rizky, H. Hastuti, and N. T. F. Anggiani, “Application of DeLone and McLean Model to Measure the Success of Institution Level Financial Application System (SAKTI),” 2025, pp. 206–214. doi: 10.2991/978-94-6463-938-4\_25.
- [63] A. I. Ojo, “Validation of the delone and mclean information systems success model,” *Healthc. Inform. Res.*, vol. 23, no. 1, pp. 60–66, Jan. 2017, doi: 10.4258/hir.2017.23.1.60.
- [64] Hernández-Sampieri, Fernández-Collado, and Baptista-Lucio, “Metodología de la investigación,” vol. 6, Apr. 2014, Accessed: Feb. 26, 2026. [Online]. Available:  
[https://apiperiodico.jalisco.gob.mx/api/sites/periodicooficial.jalisco.gob.mx/files/metodologia\\_de\\_la\\_investigacion\\_-\\_roberto\\_hernandez\\_sampieri.pdf](https://apiperiodico.jalisco.gob.mx/api/sites/periodicooficial.jalisco.gob.mx/files/metodologia_de_la_investigacion_-_roberto_hernandez_sampieri.pdf)
- [65] “IBM SPSS Statistics,” Ibm.com. Accessed: Jan. 11, 2026. [Online]. Available: <https://www.ibm.com/es-es/products/spss-statistics>
- [66] M. F. . Triola and L. Esther. Pineda Ayala, *Estadística*. Pearson Educación, 2009.
- [67] A. Matas, “Diseño del formato de escalas tipo Likert: Un estado de la cuestión,” *Revista Electronica de Investigacion Educativa*, vol. 20, no. 1, pp. 38–47, 2018, doi: 10.24320/redie.2018.20.1.1347.

## Anexos

### 6.1 Anexo A. Fotografías de la encuesta Linkert usando el modelo DeLone and McLean



## 6.2 Anexo E. Certificado de Entrega de la Aplicación Web (Organización UNORICO SAMASHUNCHIK)



UNIÓN DE ORGANIZACIONES INDÍGENAS DEL CANTÓN OTAVALO  
Acuerdo Nro. SDH-DRNPOR-2020-0111-A

**UNORICO SAMASHUNCHIK**  
(Unión de Organizaciones Indígenas del cantón Otavalo)  
Otavalo, Imbabura, Ecuador  
FECHA: 13 de febrero del 2026

**Ing. Ortega Bustamante Cosme MacArthur Msc.**  
COORDINADOR DE LA CARRERA DE INGENIERÍA DE SOFTWARE  
UNIVERSIDAD TÉCNICA DEL NORTE

Por medio del presente, en mi calidad de **Presidenta** de la organización **UNORICO SAMASHUNCHIK**, certifico que la estudiante **Verónica Cecilia Saransig Albacando**, portador de la cédula de ciudadanía N.º 1050193265 ha culminado satisfactoriamente el proceso de desarrollo, implantación y entrega del sistema informático correspondiente a su trabajo de titulación.

El software entregado consiste en una **Aplicación Web para la Gestión del Cementerio Indígena de Otavalo**, diseñado específicamente para nuestra organización. Esta herramienta ha sido desplegada en nuestros entornos operativos y puesta en funcionamiento para optimizar la administración de los difuntos, espacios y servicios del cementerio. Durante el proceso, el tesista realizó la instalación, configuración (incluyendo la gestión de base de datos y el despliegue de la aplicación en la nube) y las pruebas respectivas, verificándose el correcto cumplimiento de los requerimientos funcionales y no funcionales previamente establecidos por la organización. En virtud de lo expuesto, la organización recibe a entera satisfacción el software desarrollado, confirmando que el sistema se encuentra operativo y constituye una herramienta de apoyo significativa para la modernización y preservación de la información de nuestro Cementerio Indígena.

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

Sin otro particular, suscribo el presente.

Atentamente,



**Blanca Marisol Cotacachi Amaguaña**  
PRESIDENTA/E DE UNORICO SAMASHUNCHIK

06) 2-927-663  
unoricocosamashunchik@gmail.com  
Calle Las Almas y Bolívar