

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
ESCUELA HÁBITAT, INFRAESTRUCTURA Y CREATIVIDAD**

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

**SISTEMA WEB PARA LA GESTIÓN DE LOS PROYECTOS DE VOLUNTARIADO
DE LA DIRECCIÓN DE IDENTIDAD Y MISIÓN DE LA PUCE IBARRA**

GUANDINANGO CONEJO DAVID RUMIÑAHUI

TUTOR: PUSDÁ CHULDE SEGUNDO ELICEO

**IBARRA – ECUADOR
JULIO, 2026**

Ibarra, 22 de julio de 2026

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de integración curricular titulado: SISTEMA WEB PARA LA GESTIÓN DE LOS PROYECTOS DE VOLUNTARIADO DE LA DIRECCIÓN DE IDENTIDAD Y MISIÓN DE LA PUCE IBARRA, presentado por el estudiante GUANDINANGO CONEJO DAVID RUMIÑAHUI con cédula de ciudadanía N° 1005040447, para obtener el Título de Ingeniero en Tecnologías de la información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.

22/7/26, 5:22 p.m. Turnitin - Informe de Originalidad - SISTEMA WEB PARA LA GESTIÓN DE LOS PROYECTOS DE VOLUNTARIADO DE LA ...

Turnitin Informe de Originalidad								
Procesado el: 22-jul-2026 17:19 -05 Identificador: 3001928745 Número de palabras: 24713 Entregado: 1	<table border="1"><thead><tr><th>Índice de similitud</th><th>Similitud según fuente</th></tr></thead><tbody><tr><td rowspan="4">4%</td><td>Fuentes de Internet: 4%</td></tr><tr><td>Publicaciones: 0%</td></tr><tr><td>Trabajos: 0%</td></tr><tr><td>del estudiante: 0%</td></tr></tbody></table>	Índice de similitud	Similitud según fuente	4%	Fuentes de Internet: 4%	Publicaciones: 0%	Trabajos: 0%	del estudiante: 0%
Índice de similitud	Similitud según fuente							
4%	Fuentes de Internet: 4%							
	Publicaciones: 0%							
	Trabajos: 0%							
	del estudiante: 0%							
SISTEMA WEB PARA LA GESTIÓN DE LOS PROYECTOS DE VOLUNTARIADO DE LA DIRECCIÓN DE IDENTIDAD Y MISIÓN DE LA PUCE IBARRA Por David Rumiñahui GUANDINANGO CONEJO								
Coincidencia del 2% (Internet desde 11-sept-2025) https://repositorio.puce.edu.ec/server/api/core/bitstreams/ac7b0a60-6c27-4f24-919e-c377e818d5ce/content								
Coincidencia del 1% (Internet desde 18-mar-2025) https://repositorio.puce.edu.ec/server/api/core/bitstreams/d2636891-2e97-421e-ae8a-832d48a77111/content								



Validar únicamente en FirmaEC.
Firmado electrónicamente por:
SEGUNDO ELICEO
PUSDÁ CHULDE

(f):
Mgs. PUSDÁ Chulde Segundo Eliceo
TUTOR DE TRABAJO
C.C.: 0401567938

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:

(f):  Validar únicamente en FirmaBC.
Firmado electrónicamente por:
SEGUNDO ELICEO
PUSDA CHULDE

Mgs. Pusdá Chulde Segundo Eliceo

C.C.: 0401567938

GALO
HERNAN
PUETATE
HUERA
(f):  Firmado digitalmente por
GALO HERNAN
PUETATE HUERA
Fecha: 2026.07.23
08:06:31 -05'00'

Msc. Galo Hernán Puetate Huera

C.C.: 0401375787

(f):  Firmado electrónicamente por:
JORGE JEFFREY
VIVERO GARCIA
Fecha: 2026-07-23T13:26:58Z
Nexus Soluciones

Mgs. Jorge Jeffrey Vivero García

C.C.: 1002061420

ACTA DE CESIÓN DE DERECHOS

Yo, *Guandinango Conejo David Rumiñahui*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilidades de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 22 de julio de 2026

David
Rumiñahui
Guandinango
(f): Conejo

Firmado
digitalmente por
David Rumiñahui
Guandinango Conejo
Fecha: 2026.07.22
20:07:09 -05'00'

Guandinango Conejo David Rumiñahui
C.C.: 1005040447

AUTORIA

Yo, Guandinango Conejo David Rumiñahui, portador de la cédula de ciudadanía N°1005040447, declaro que el presente trabajo de investigación es de total responsabilidad del autor, y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.

David
Rumiñahui
Guandinango
(f): Conejo

Firmado digitalmente
por David Rumiñahui
Guandinango Conejo
Fecha: 2026.07.22
20:07:31 -05'00'

Guandinango Conejo David Rumiñahui

C.C.: 1005040447

DEDICATORIA

A Dios, por guiar cada uno de mis pasos y darme la fortaleza para alcanzar esta meta, y a la Santísima Virgen María, por cubrirme siempre con tu santo manto y acompañarme en cada momento de este camino.

A mis padres, Ángel Guandinango y María Espíritu Conejo, por ser el pilar de mi vida, mi mayor inspiración y el motivo para nunca rendirme, gracias por cada sacrificio, por cada esfuerzo silencioso, por creer en mí incluso cuando el camino parecía difícil, y por enseñarme que el trabajo, la humildad y la perseverancia siempre tienen su recompensa, pero sobre todo, por enseñarme a ser, ante todo, un buen ser humano; todo lo que soy y este logro también les pertenece. A mis hermanas, Toa, Tamia, Mayra y Judith, por ser un ejemplo para mí, por su cariño, su amor y por alentarme a seguir adelante en cada etapa de este camino.

A mi querida Janamamita y a mis abuelitos, Juana y Antonio, quienes desde el cielo siguen iluminando mi camino y cuidando de mí, y a ti, mi hermano Inti, por acompañarme desde el cielo y por darme fuerzas para seguir adelante. Lo logramos.

AGRADECIMIENTOS

A la Dirección de Identidad y Misión de la PUCE Ibarra y, de manera especial, a su directora, por la confianza, el apoyo y la apertura brindados durante el desarrollo de este trabajo. Mi sincero agradecimiento a la PhD. Laura Guerra y a los Mgs. José Luis Ibarra, Segundo Puská, Jorge Vivero y Galo Puetate, quienes, con sus enseñanzas, experiencia y dedicación, dejaron una huella importante en mi formación profesional. Finalmente, agradezco a todas las personas que, de una u otra manera, me acompañaron y apoyaron durante este proceso, contribuyendo a que este logro fuera posible.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN TUTOR.....	ii
PÁGINA DE APROBACIÓN DEL TRIBUNAL	iii
ACTA DE CESIÓN DE DERECHOS.....	iv
AUTORIA.....	v
DEDICATORIA.....	vi
AGRADECIMIENTOS	vii
ÍNDICE DE CONTENIDOS.....	viii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS	xii
RESUMEN	xiv
ABSTRACT	xv
INTRODUCCIÓN	xvi
OBJETIVO GENERAL.....	xviii
OBJETIVOS ESPECÍFICOS	xviii
CAPÍTULO I. ESTADO DEL ARTE.....	1
1.1. Descripción de la empresa	1
1.2. Investigaciones previas	2
1.2.1. Herramienta para la gestión y coordinación del voluntariado	2
1.2.2. Sistema informático para la gestión de voluntarios y administración de proyectos de Techo el salvador.....	3
1.2.3. Sistema informático para la gestión de los voluntariados del UCC	4
1.3. Conceptos básicos.....	5
1.3.1 Aplicación web	5
1.3.2. Arquitecturas de software	6
1.3.3. API REST.....	8
1.3.4. Escalabilidad en PostgreSQL	8
1.3.5. Metodologías ágiles de desarrollo	9
1.3.6. Diseño UX/UI: Adopción y satisfacción del usuario	10
1.3.7. Diferenciación técnica entre UX y UI	11
1.3.8. Gestión de proyectos y eficiencia organizacional (Soluciones informáticas existentes para la gestión de proyectos).....	11

1.3.9. Seguridad en aplicaciones web	12
1.3.10. Frameworks Frontend	14
1.3.11. Frameworks Backend	14
1.3.12. Lenguaje de modelado unificado (uml)	15
1.3.13. Pruebas de software	16
CAPÍTULO II. MATERIALES Y MÉTODOS	18
2.1. Aspectos generales de la investigación	18
2.1.1. Propósito	19
2.1.2. Alcance	19
2.2. Metodología de la propuesta tecnológica	20
2.2.1. Planificación del Product Backlog	21
2.2.2. Fase de diseño	29
2.2.3. Fase de codificación	39
2.2.4. Plan de pruebas	42
CAPÍTULO III. RESULTADOS Y DISCUSIÓN	48
3.1. Interfaces del sistema	48
3.1.1. Módulo de administración del voluntariado (Coordinador)	49
3.1.2. Módulo de gestión de proyectos de voluntariado (Líder)	57
3.1.3. Módulo de reportes del voluntariado	63
3.2. Validación del sistema (pruebas funcionales)	65
3.2.1. Resumen general de resultados	70
3.2.2. Análisis de hallazgos y observaciones	71
3.3. Discusión de resultados	71
3.3.1. Comparación con trabajos previos	72
3.3.2. Interpretación de los resultados	73
3.3.3. Limitaciones del estudio	73
3.3.4. Consideraciones sobre los aspectos no funcionales	74
3.3.5. Implicaciones prácticas y organizacionales	75
CONCLUSIONES	76
RECOMENDACIONES	77
REFERENCIAS BIBLIOGRÁFICAS	79
ANEXOS	85
Anexo 1. Guion de entrevista	85
Anexo 2. Ficha de Planificación de Actividades de voluntariado	86

Anexo 3. Carta de recepción de la propuesta tecnológica.....	87
--	----

ÍNDICE DE TABLAS

Tabla 1	Tabla comparativa de las metodologías ágiles de desarrollo.....	10
Tabla 2	Tabla comparativa de las herramientas de gestión de proyectos	12
Tabla 3	Tipos de diagramas Existentes del Lenguaje de Modelado Unificado (UML)	15
Tabla 4	Historias de usuario del sistema web de voluntariado	23
Tabla 5	Tabla de análisis de las historias de usuario	27
Tabla 6	Distribución de historias de usuario en Sprints.....	29
Tabla 7	Plantilla del Plan de Pruebas del Sistema Web de voluntariado.....	43
Tabla 8	Plantilla Prueba funcional — Inicio de sesión del usuario	43
Tabla 9	Plantilla Prueba funcional — Gestión de proyectos de voluntariado	44
Tabla 10	Plantilla Prueba funcional — Solicitud de ejecución del proyecto.....	44
Tabla 11	Plantilla Prueba funcional — Registro de participantes voluntarios	45
Tabla 12	Plantilla Prueba funcional — Planificación de actividades del proyecto	45
Tabla 13	Plantilla Prueba funcional — Autorización del coordinador.....	46
Tabla 14	Plantilla Prueba funcional — Registro de actividades realizadas y evidencias.....	46
Tabla 15	Plantilla Prueba funcional — Cierre del proyecto e informe de ejecución.....	47
Tabla 16	Prueba unitaria: inicio de sesión	65
Tabla 17	Prueba unitaria: gestión de proyectos de voluntariado	66
Tabla 18	Prueba unitaria: solicitud de ejecución del proyecto	66
Tabla 19	Prueba unitaria: registro de participantes voluntarios.....	67
Tabla 20	Prueba unitaria: planificación de actividades del proyecto.....	67
Tabla 21	Prueba unitaria: autorización del cambio de estado.....	68
Tabla 22	Prueba unitaria: registro de evidencias de ejecución	68
Tabla 23	Prueba unitaria: cierre del proyecto e informe final	69
Tabla 24	Prueba unitaria: reportes del voluntariado	69
Tabla 25	Resumen de resultados de la validación funcional	70

ÍNDICE DE FIGURAS

Figura 1	Diagrama de la arquitectura del sistema web de la Plataforma Coruñesa de Voluntariado	3
Figura 2	Coste de resolución de fallos de la fase del ciclo de vida donde se detecta.	13
Figura 3	Marco de trabajo Scrum	21
Figura 4	Diagrama de procesos del sistema de voluntariado.....	30
Figura 5	Diagrama de casos de uso Administración del Voluntariado.....	31
Figura 6	Diagrama de casos de uso Módulo de Seguimiento de Proyectos de Voluntariado (Líder)	32
Figura 7	Diagrama de casos de uso Módulo de reportes del voluntariado	33
Figura 8	Diagrama de escenario del proceso de autenticación.....	34
Figura 9	Diagrama de Arquitectura en capas del sistema web	35
Figura 10	Diagrama de Base de Datos Voluntariado	37
Figura 11	Prototipo de visualización responsiva.	38
Figura 12	Inicio de sesión del sistema de voluntariado	49
Figura 13	Gestión de áreas de voluntariado y registro de una nueva área.....	50
Figura 14	Listado de beneficiarios.....	50
Figura 15	Registro de un beneficiario.....	51
Figura 16	Gestión de proyectos de voluntariado con filtros por estado	51
Figura 17	Registro de un nuevo proyecto	52
Figura 18	Proyecto en estado Creado	53
Figura 19	Asignación del líder responsable.....	53
Figura 20	Proyecto enviado a revisión con las acciones de decisión	54
Figura 21	Revisión de la planificación enviada.....	54
Figura 22	Rechazo de la planificación con registro del motivo	55
Figura 23	Proyecto reenviado con la decisión pendiente	55
Figura 24	Revisión de participantes del proyecto por el coordinador	56
Figura 25	Consulta de evidencias de ejecución por el coordinador	56
Figura 26	Panel del líder con un proyecto asignado.....	57
Figura 27	Registro de un participante del proyecto.....	58
Figura 28	Listado de participantes presentes.....	58
Figura 29	Historial de participantes eliminados	59
Figura 30	Registro de la planificación del proyecto	59

Figura 31 Actividades registradas de la planificación	60
Figura 32 Envío de la planificación a revisión	60
Figura 33 Planificación rechazada con el motivo visible y opción de reenvío.....	61
Figura 34 Registro de evidencias de ejecución en formato PDF	61
Figura 35 Finalización del proyecto con informe final obligatorio	62
Figura 36 Proyecto finalizado con acceso al informe final	62
Figura 37 Tablero de reportes con indicadores clave (KPI)	63
Figura 38 Reportes de proyectos generados por año y participantes por proyecto	64
Figura 39 Reportes de beneficiarios y actividades por período	64

RESUMEN

En la PUCE Ibarra, la Dirección de Identidad y Misión tiene a su cargo la formación humanística de la comunidad universitaria, que se promueve mediante proyectos de voluntariado en distintas áreas de acción. Al gestionarse estos proyectos de manera manual, la dirección no disponía de información centralizada sobre las iniciativas vigentes ni sobre sus participantes, y la obtención de reportes e indicadores obligaba a la coordinación a revisar individualmente las fichas que remitía cada responsable, información que además solo se consolidaba una vez concluido el proyecto y quedaba expuesta a extravío. Ante ello, la investigación se propuso desarrollar un sistema web destinado a la gestión de los proyectos de voluntariado de esta dirección, mediante frameworks basados en TypeScript. La solución distingue las responsabilidades del coordinador y del líder de proyecto, y estructura el recorrido de cada iniciativa mediante estados que abarcan su creación, la designación del responsable, la aprobación, la ejecución con respaldo de evidencias y su cierre; dispone asimismo de un módulo de reportes que genera indicadores y gráficos estadísticos de forma automática. Su construcción se sustentó en Scrum, marco que permitió distribuir las historias de usuario en sprints y liberar incrementos funcionales sucesivos. Se recurrió a tecnologías de vanguardia como Nuxt, NestJS y PostgreSQL, complementadas con Git para el control de versiones y Swagger para documentar los servicios. Las pruebas funcionales resultaron aprobadas en la totalidad de los escenarios verificados, confirmando que el sistema reúne en un solo entorno los proyectos, sus participantes y sus evidencias. Se concluye que la solución permite disponer de información centralizada de los proyectos durante su ejecución y no solo al concluirlos, superando las limitaciones detectadas en el diagnóstico inicial y consolidándose como una herramienta base para la transformación digital de la gestión del voluntariado universitario.

Palabras clave: gestión de voluntariado, sistema web, PUCE-I, Scrum, Nuxt, NestJS, PostgreSQL

ABSTRACT

At PUCE Ibarra, the Directorate of Identity and Mission is responsible for the humanistic formation of the university community, which is promoted through volunteering projects in different areas of action. Because these projects were managed manually, the directorate did not have centralized information about ongoing initiatives or about who was taking part in each one, and obtaining reports and indicators required the coordination team to individually review the record sheets submitted by each person in charge — information that was moreover only consolidated once the project had ended and remained exposed to loss. In view of this, the research set out to develop a web system for managing the volunteering projects of this directorate, using frameworks based on TypeScript. The solution distinguishes the responsibilities of the coordinator and the project leader, and structures the course of each initiative through states covering its creation, the appointment of the person in charge, approval, execution supported by evidence, and its closure; it also provides a reporting module that automatically generates indicators and statistical charts. Its construction was based on Scrum, a framework that made it possible to distribute user stories across sprints and release functional increments successively. State-of-the-art technologies such as Nuxt, NestJS, and PostgreSQL were used, complemented by Git for version control and Swagger for service documentation. The functional tests yielded an approved result in all verified scenarios, confirming that the system brings together projects, participants, and evidence in a single environment. It is concluded that the solution makes it possible to have centralized information about the projects during their execution, and not only upon their completion, overcoming the limitations detected in the initial diagnosis and establishing itself as a foundational tool for the digital transformation of university volunteering management.

Keywords: volunteer management, web system, PUCE-I, Scrum, Nuxt, NestJS, PostgreSQL

INTRODUCCIÓN

Durante las últimas décadas, la incorporación de las Tecnologías de la Información y Comunicación (TIC) se ha convertido en un elemento esencial para la modernización de las organizaciones en el mundo. La tecnología no solo simplifica el trabajo diario, sino que ha transformado profundamente la forma en que las instituciones administran sus servicios y toman decisiones estratégicas (Suescum et al., 2025).

Hoy en día, el desarrollo de aplicaciones web y la adopción de arquitecturas de software modernas y escalables se han convertido en pilares fundamentales para la transformación digital y la optimización de procesos en las organizaciones. Asimismo, los sistemas web de gestión centralizados y modulares permiten consolidar en una misma plataforma la información más relevante de un departamento u organismo, permitiendo una mejor operatividad. El uso de tecnologías de la información, específicamente sistemas web escalables basados en frameworks actuales, permite centralizar grandes volúmenes de datos dispersos para convertirlos en recursos claros, útiles y seguros. La integración de estas herramientas informáticas otorga una interacción fluida a los usuarios, automatiza la captura de información y suprime los errores desapercibidos que se generan por el manejo de procesos manuales, mejorando el seguimiento y la toma de decisiones institucionales (Chávarry & Sandoval, 2023).

Una de las tecnologías principales que se utilizan para el desarrollo de este tipo de sistemas web es Node.js, que permite el desarrollo de software progresivo desde el lado del servidor utilizando lenguajes de programación robustos como Typescript, permitiendo crear sistemas web mantenibles y escalables (Kyriakou & Tselikas, 2022).

La aplicación de este tipo de soluciones resulta especialmente pertinente en instituciones que requieren una gestión eficiente de múltiples procesos y actores, como ocurre en el ámbito universitario. En estos entornos, la integración de sistemas web favorece la automatización de actividades administrativas, y también fortalece el control, seguimiento y evaluación de las distintas funciones que se desarrollan dentro de una organización.

Bajo esta perspectiva, la Pontificia Universidad Católica del Ecuador Ibarra (PUCE-Ibarra) se constituye como una institución de educación superior de gran prestigio reconocida por su compromiso con la excelencia académica, la formación integral, y la mejora continua de sus procesos (PUCEI, 2025). Como estructura organizacional, la universidad promueve acciones orientadas al fortalecimiento de sus áreas académicas, administrativas y de vinculación con la sociedad. Dentro de esta estructura organizacional se encuentra una dependencia fundamental,

la Dirección de Identidad y Misión (DIM), encargada de coordinar la Identidad Institucional, los Servicios Académicos misionales y el Bienestar Universitario, por lo que se constituye en una instancia central para la vivencia y fortalecimiento del ideario católico, la formación humanista, ética y el compromiso social universitario; en este contexto, la DIM, al ser la encargada de gestionar la formación humanística de la universidad, desarrolla esta labor a través del voluntariado universitario, en el que participan docentes, estudiantes, proceso gestionado por el área de Voluntariado, que coordina diversos proyectos de acción social y voluntariado, fortaleciendo así la formación integral y el compromiso social de la comunidad universitaria.

La pertinencia de este trabajo radica, además, en su alineación con el rol social y normativo de las instituciones de educación superior. Fortalecer la gestión del voluntariado no solo optimiza un proceso administrativo, sino que respalda la formación integral y ética de la comunidad universitaria. Esta orientación es coherente con el Modelo Educativo de la PUCE, que sitúa a la persona como el centro del proceso formativo y promueve la vinculación con la sociedad como un compromiso institucional (Cevallos A. , 2025) asimismo, se articula con lo dispuesto por la Ley Orgánica de Educación Superior, cuyo artículo 107 establece que las instituciones de educación superior deben vincularse con las necesidades del entorno social (Asamblea Nacional del Ecuador, 2010) En este sentido, disponer de una herramienta que organice la información y evidencie el impacto social generado contribuye directamente a la misión formativa de la institución.

Sin embargo, al existir una gran diversidad de proyectos de voluntariado, se evidenció un problema operativo: la coordinación del voluntariado no contaba con información centralizada sobre las iniciativas vigentes. Específicamente, para la obtención de indicadores de gestión, la directora de la dirección debía analizar detalladamente y de forma manual cada hoja de las fichas físicas que le entregaba cada líder de proyecto. La consolidación de estos indicadores es de vital importancia para la institución, puesto que mediante estos datos se sustenta la evaluación de la universidad. En consecuencia, al no tener información centralizada sobre el número real de participantes o actores involucrados, se generaba un escenario de incertidumbre administrativa. Esta dispersión y el trabajo manual dificultaban el rastreo del estado de los proyectos y limitaban severamente la capacidad de la institución para tomar decisiones estratégicas.

Ante esta necesidad de unificar la información, y además considerando que la implementación de sistemas tecnológicos es fundamental para optimizar los procesos y servicios ofrecidos, el presente trabajo propuso el desarrollo de un sistema web integral para la gestión del

voluntariado universitario, con el fin de optimizar el flujo de trabajo entre coordinadores, líderes y participantes. De esta manera, se buscó dar respuesta a la siguiente pregunta de investigación: ¿Qué funcionalidades debe cumplir un sistema web de gestión de voluntariado para mejorar la trazabilidad y seguimiento de los proyectos gestionados por la Dirección de Identidad y Misión de la PUCE-Ibarra?

Para dar respuesta a esta interrogante y establecer la hoja de ruta metodológica y técnica del proyecto, se definieron los siguientes objetivos:

OBJETIVO GENERAL

Desarrollar un sistema web para la gestión de los proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra, mediante frameworks basados en TypeScript.

OBJETIVOS ESPECÍFICOS

- Analizar el proceso actual de gestión de los proyectos de voluntariado en la Dirección de Identidad y Misión identificando las limitaciones y los puntos críticos en el seguimiento y control de la información de proyectos.
- Definir aquellos requisitos funcionales y no funcionales previo al desarrollo de la solución tecnológica, el sistema web de voluntariado, de acuerdo con las necesidades gestoras y operativas de la PUCE Ibarra.
- Diseñar la arquitectura del sistema web teniendo en cuenta la consideración de la gestión de roles de usuario.
- Validar el sistema web mediante pruebas de funcionalidades para verificar el cumplimiento de los requisitos.

Para alcanzar estos objetivos y desarrollar la investigación de manera sistemática, el presente documento se estructura en tres capítulos: El primero aborda el estado del arte, los antecedentes y la fundamentación conceptual que sustentan el desarrollo de la aplicación y la gestión del voluntariado. El segundo capítulo se centra en la metodología del proyecto, el análisis de requisitos y el diseño de la arquitectura y de interfaces (UX/UI) de la plataforma. Finalmente, el tercer capítulo presenta el desarrollo del sistema web con sus respectivos módulos para validar su funcionamiento.

CAPÍTULO I. ESTADO DEL ARTE

Este capítulo constituye el fundamento teórico y referencial que sustenta el desarrollo de la presente investigación. Se presentan antecedentes relacionados con la Dirección de Identidad y Misión de la PUCE Ibarra, con el propósito de comprender su estructura y el rol que desempeña el área de voluntariado dentro de la institución. Además, se aborda el marco conceptual que sustenta el desarrollo del sistema web, incluyendo temas como voluntariado universitario, gestión de proyectos, tecnologías web, frameworks y metodologías de desarrollo de software. Por último, se revisan antecedentes y trabajos relacionados con soluciones tecnológicas para la administración de proyectos sociales, los cuales contribuyeron a identificar los elementos para el desarrollo de la propuesta.

1.1. Descripción de la empresa

El voluntariado universitario en la Pontificia Universidad Católica del Ecuador, Sede Ibarra, se ha consolidado como una estrategia fundamental para vincular la formación académica con la realidad social. A través de la Dirección de Identidad y Misión, en coordinación con el área de Identidad Institucional, se desarrollan iniciativas orientadas a fortalecer el compromiso social de los estudiantes mediante su participación en proyectos comunitarios.

En estos programas de voluntariado participan tanto líderes y voluntarios. Bajo esta perspectiva, se busca formar personas con valores éticos de carácter Ignaciano e Idente, promoviendo el compromiso comunitario y generando un impacto positivo en sectores vulnerables. De esta manera, se articulan procesos de formación académica, humana y social, contribuyendo al bienestar universitario, la igualdad de oportunidades, el desarrollo de la responsabilidad y el fortalecimiento de la interculturalidad (Cevallos A. , 2025).

Las iniciativas de voluntariado abarcan diversas áreas de intervención social, lo que evidencia la amplitud y complejidad de los proyectos gestionados. Esta labor implica la participación de múltiples actores (coordinador, líderes, voluntarios) y el manejo de información relacionada con estudiantes, docentes y procesos organizativos.

En la actualidad, este proceso inicia cuando el coordinador de voluntariado asigna los proyectos de voluntariado a los líderes. El líder es quien selecciona a los participantes del proyecto de voluntariado al cual ha sido asignado, que pueden ser estudiantes o graduados. Posteriormente, el líder docente elabora un documento de planificación en el que se detallan las actividades a realizar, los beneficiarios y los recursos necesarios para el proyecto, el cual es aprobado por la directora de la Dirección de Identidad y Misión (véase Anexo 2). Una vez definidos el número

de participantes y los aspectos sin conflictos, se inicia el proyecto de voluntariado para su posterior ejecución. En la fase de finalización, se procede a evaluar el seguimiento realizado durante el proyecto. Para ello, el líder docente debe llenar una ficha con ciertos parámetros, utilizando métodos manuales y tradicionales para documentar la información. Luego, las fichas se envían mediante medios no centralizados para llegar a la Dirección de Identidad y Misión. Al recibir los documentos, la coordinación los analiza manualmente, revisando cada ficha entregada por los líderes.

Por consiguiente, una adecuada gestión del voluntariado universitario es clave para que los proyectos de acción social y vinculación con la sociedad funcionen bien y alcancen sus objetivos. Como señala Yelin (2025), cuando esta gestión se adapta a las necesidades reales, motiva a los voluntarios a crecer personal y profesionalmente, generando un sentido de satisfacción y utilidad social. Asimismo, los participantes encuentran gratificación emocional y personal al colaborar con la colectividad.

1.2. Investigaciones previas

1.2.1. Herramienta para la gestión y coordinación del voluntariado

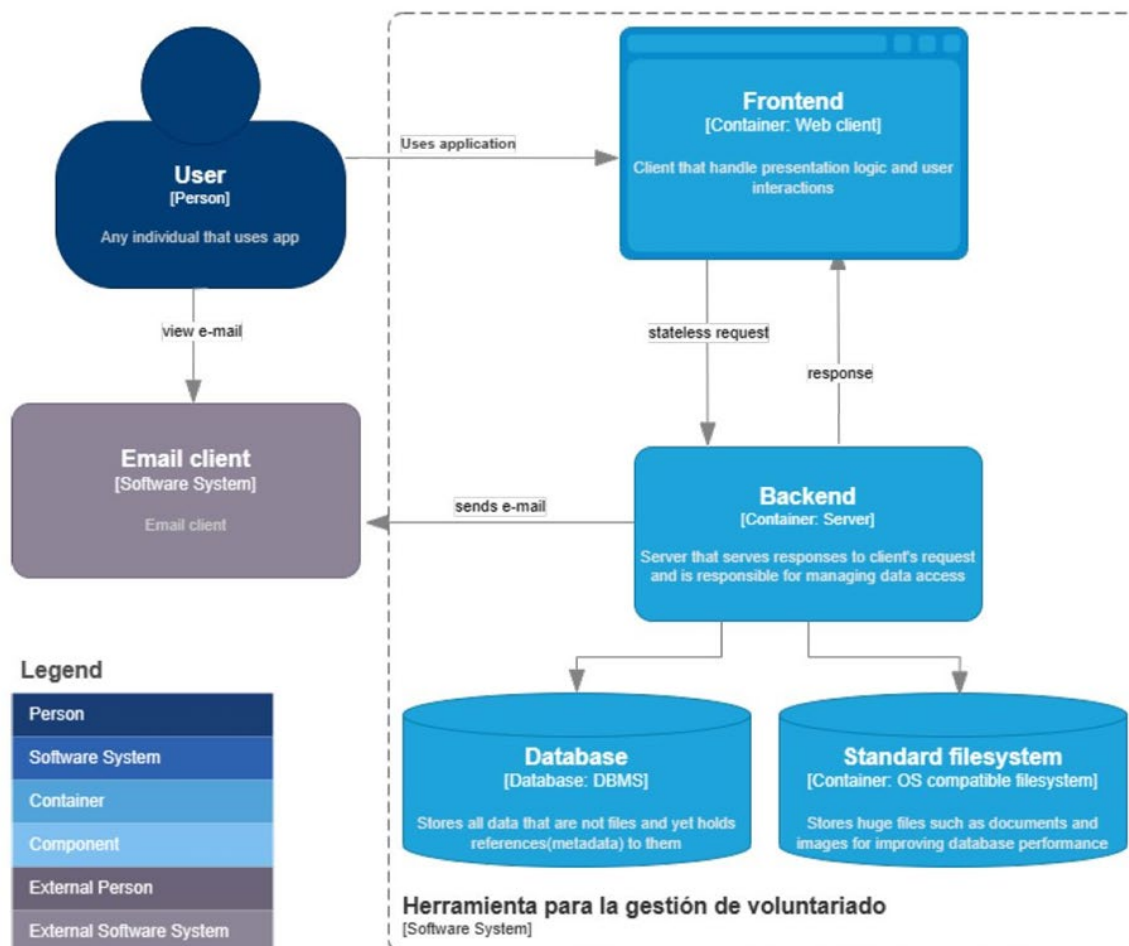
Fraguío (2023) desarrolló una “Plataforma web de código abierto para la gestión y coordinación del voluntariado” para la Plataforma Coruñesa de Voluntariado, mediante una arquitectura cliente-servidor organizada en capas de lógica de negocio, acceso a datos y servicios REST. Para la construcción del sistema, se empleó un framework de Java en el backend, React con Bootstrap en el frontend y PostgreSQL para la base de datos. La administración de los puntos de conexión (endpoints) se realizó mediante Swagger, facilitando la comunicación entre las capas del software. Además, se diseñaron los prototipos de las pantallas bajo metodologías UX/UI y se integró la clasificación de proyectos según los Objetivos de Desarrollo Sostenible (ODS). Los resultados indican que esta solución permite centralizar la información y generar indicadores, cumpliendo con los requerimientos de trazabilidad de los voluntarios.

La arquitectura de software empleada en el prototipado de este sistema web por parte del autor se puede evidenciar en la Figura 1, donde se aprecia la separación entre el cliente web (frontend), el servidor en capas (backend) y los sistemas de almacenamiento externos. Esta organización arquitectónica constituye un referente para la presente investigación, particularmente en lo que respecta a la estructuración en capas independientes y la

comunicación sin estado entre cliente y servidor, principios que orientan las decisiones de diseño del sistema propuesto.

Figura 1

Diagrama de la arquitectura del sistema web de la Plataforma Coruñesa de Voluntariado



Nota. Tomado de "Plataforma web de código abierto para la gestión y coordinación del voluntariado", por Fraguío Sánchez, J. (2023). Herramienta para la gestión y coordinación del voluntariado. Universidad de la Coruña. Derechos reservados.

1.2.2. Sistema informático para la gestión de voluntarios y administración de proyectos de Techo el salvador

En una línea similar de optimización operativa, Aragón et al., (2022) analizaron un sistema orientado a mejorar la gestión administrativa en TECHO El Salvador, una organización que

moviliza cada año entre 800 y 1000 voluntarios. Bajo un modelo de ciclo de vida en cascada, esta investigación se centró en la fase de análisis y diseño de sistemas, utilizando técnicas de recolección de información como entrevistas y encuestas para identificar las necesidades institucionales. Se estructuró una solución mediante el modelado de procesos con UML y definió los requerimientos funcionales para la evaluación de desempeño y el seguimiento de actividades. El diseño técnico se orientó al uso de Angular en el frontend, Express en el backend y MySQL como base de datos, estableciendo una base para la centralización de datos y el manejo de indicadores de gestión en proyectos de vinculación social. Como resultado de este proceso, se definen módulos para la gestión de expedientes digitales clasificados por perfiles y un componente para el control de horas de servicio social universitario mediante bitácoras digitales. El estudio demuestra que utilizar una metodología para desarrollo de software es primordial; sin embargo, el alcance del software se limitó a la gestión operativa de procesos, sin integrar funciones para la generación de reportes estadísticos automáticos sobre la situación técnica de los proyectos (Aragón et al., 2022).

Esta referencia proporciona una base metodológica al emplear el modelo en cascada y herramientas como UML para el análisis y diseño de sistemas, lo cual sirve como referencia para estructurar el desarrollo del sistema propuesto.

1.2.3. Sistema informático para la gestión de los voluntariados del UCC

Asimismo, se desarrolló un "Sistema Informático para la Gestión de los Voluntariados del UCC". Utilizando una metodología de desarrollo colaborativa e iterativa, el autor analizó la informalidad de los registros y la descentralización administrativa, y construyó el sistema informático web empleando Python, el framework Django, Wagtail y PostgreSQL; así se logró la automatización de procesos operativos, permitiendo que los voluntarios gestionen su propia asistencia y traslados a los puntos de encuentro, lo que redujo significativamente la carga de trabajo manual y mejoró la transparencia de la información institucional (Bobadilla, 2024). Los resultados ratifican que el proyecto del UCC demandó una investigación constante en áreas de codificación, diseño de base de datos y despliegue, donde la validación y el diálogo con el cliente permiten consolidar una solución efectiva con un impacto social positivo en la gestión de los voluntariados universitarios. El enfoque de diseño está centrado en el usuario, lo que asegura que cada módulo del sistema se adapte a las necesidades y responsabilidades de cada tipo de usuario. Además, se encuentra alineado con la política de Responsabilidad Social Universitaria (RSU) de la Universidad Católica de Córdoba, destacando el impacto

organizacional y social del sistema. Este enfoque puede servir como modelo para la investigación de ingeniería de software, el cual busca generar un impacto positivo en la sociedad.

Esta referencia resulta particularmente valiosa para la investigación, ya que proporciona lineamientos claros para la gestión de roles y permisos, así como para la definición de vistas específicas según el rol de cada usuario dentro del sistema. En síntesis, el trabajo de Bobadilla (2024) constituye un aporte técnico y conceptual fundamental para esta investigación. Desde el punto de vista práctico, el acceso a la arquitectura del proyecto a través de su repositorio en GitHub ofrece recursos invaluable para el análisis y desarrollo, sirviendo como guía para el diseño de esquemas de roles, permisos y vistas personalizadas.

De este modo, las investigaciones analizadas evidencian que la incorporación de plataformas web es un factor determinante para reducir la carga administrativa y mejorar la transparencia de la gestión de proyectos de voluntariado. Los trabajos revisados demuestran que el éxito de estas soluciones radica en la selección de arquitecturas robustas y metodologías ágiles de desarrollo que garanticen la escalabilidad, asegurando la integridad de los datos y la generación de información para mejor toma de decisiones.

1.3. Conceptos básicos

1.3.1 Aplicación web

Las aplicaciones web son soluciones de software que requieren una red para su funcionamiento y acceso. Generalmente, están compuestas por una capa de presentación, denominada frontend, que permite la interacción con el usuario, y por una capa lógica o de cálculos, conocida como backend, encargada de gestionar los datos y ejecutar las funcionalidades del sistema (Celi et al., 2023).

Para su diseño y desarrollo, se requiere el uso de tecnologías con un rendimiento adecuado, especialmente en el backend, debido a que en esta capa recaen procesos como la gestión de conexiones concurrentes, las operaciones de entrada y salida, y la ejecución de tareas intensivas en procesamiento que incrementan la carga sobre el CPU del servidor donde se ejecuta la aplicación, factores que inciden en la escalabilidad y eficiencia del sistema (Khokhriagkov, 2024).

1.3.2. Arquitecturas de software

La arquitectura monolítica. Es el método de desarrollo de software en el que todos los elementos de una aplicación (interfaz, lógica empresarial y acceso a datos) se combinan en una sola estructura de codificación y se implementan en forma de un solo conjunto (Blinowski et al., 2022). La revisión sistemática sobre migración desde arquitecturas monolíticas hacia microservicios, describen este modelo como aquel donde todos los procesos están fuertemente acoplados y se ejecutan como un servicio unificado, lo que simplifica su desarrollo inicial y reduce los costos de despliegue en etapas tempranas del proyecto (Hassan et al., 2024).

Arquitectura de capas. La arquitectura de tres capas es otra forma de estructurar el desarrollo de software, donde el sistema se divide en la interfaz de presentación, responsable de la interacción humano-computadora con el usuario; la capa abstracta de negocio, donde se procesan las reglas y compilación del dominio; y la capa de acceso, encargada del almacenamiento y recuperación de información. Esta separación de responsabilidades permite procesar información en paralelo, ya que una modificación en una capa tiene efecto mínimo sobre las demás (Tu, 2023).

Arquitectura modelo vista controlador (MVC). Dentro de estos enfoques que materializan los principios de la arquitectura de un software a nivel de implementación, el patrón Modelo-Vista-Controlador (MVC) representa una de las soluciones más consolidadas en el desarrollo de aplicaciones web, al permitir una separación clara entre la capa de datos, la capa de presentación y la lógica de interacción con el usuario (Khan & Khanam, 2023). Este patrón organiza el sistema en tres componentes: el Modelo, que gestiona la lógica de negocio y el acceso a datos; la Vista, encargada de la presentación de información al usuario; y el Controlador, que coordina las interacciones entre ambas capas procesando las solicitudes entrantes (Necula, 2024). En su análisis sobre la arquitectura MVC y sus aplicaciones tecnológicas, Necula concluye que este patrón es ampliamente utilizado tanto en desarrollo web como en soluciones empresariales, precisamente porque su estructura facilita el mantenimiento, promueve la reutilización de código y reduce el acoplamiento entre componentes, lo cual representa una enorme diferencia respecto a los software monolíticos tradicionales, lo que se traduce en aplicaciones más organizadas, eficientes y sostenibles en el tiempo.

De la misma manera Guntakandla (2025) dice que, a diferencia de los sistemas tradicionales, específicamente de los monolíticos, donde la estructura impide que cada unidad funcione de forma aislada y que se pueda escalar de manera independiente —provocando que los cambios

en una parte requieran ajustes en otras partes del sistema—, también existe la idea de modularidad que sí permite la conexión entre los componentes de manera más organizada y controlada.

La arquitectura modular. Es un paradigma de diseño de software que organiza los sistemas informáticos en un conjunto de componentes independientes y funcionales. Este enfoque consiste en descomponer un sistema en partes autónomas que se comunican mediante elementos que se abstraen individualmente y ordenan su interacción. Es así que, la arquitectura basada en módulos se sustenta en principios como la encapsulación, donde cada módulo protege su lógica interna exponiendo solo lo necesario, y el bajo acoplamiento que minimiza las dependencias, facilitando la sustitución o actualización de componentes sin comprometer la estabilidad global del sistema (Guntakandla, 2025).

Monolito modular. La integración de software basada en módulos constituye uno de los desafíos centrales en el diseño de aplicativos escalables tal y como lo dice (Barde, 2023). Sin embargo, el mismo defiende que el patrón arquitectónico denominado Monolito Modular ha emergido como una solución intermedia para los sistemas microservicios y los monolíticos tradicionales, buscando aprovechar las ventajas de ambos enfoques sin incurrir en sus respectivos costos.

Volviendo a la idea de Guntakandla (2025) en un monolito tradicional, las partes (controladores, servicios y modelos) se integran mediante dependencias directas en una misma capa técnica, lo que genera un alto acoplamiento inter-modular y dificulta el mantenimiento evolutivo del sistema. En contraste, el Monolito Modular conecta estos paradigmas y organiza el código adaptándose a la lógica de negocio, donde cada módulo encapsula su propia lógica, expone interfaces bien definidas y se comunica con otros módulos a través de mecanismos de inyección de dependencias o eventos internos, reduciendo el acoplamiento estructural (Su & Li, 2024)

El enfoque de integración de módulos en un proyecto de e-learning y sus resultados empíricos demuestran que esta forma de integración modular mejora significativamente la mantenibilidad del sistema elevando la calificación de SonarQube de grado C a grado A y reduciendo el índice de deuda técnica de 11.5% a 4.5% sin el costo operativo que implica una arquitectura de microservicios completa. Esta forma de desarrollo es una alternativa a los microservicios y que también podría constituir un paso previo antes de que los sistemas comunes migren a los microservicios (Arroziqi & Susanty, 2026).

El éxito de esta integración reside en la autonomía de los módulos federados y el uso de TypeScript para garantizar la seguridad de tipos, tecnología que asegura a una plataforma

evolucionar según las demandas institucionales sin alterar el núcleo del software, optimizando la administración centralizada de dependencias compartidas y previniendo fallas durante el acoplamiento de nuevas piezas (Lando & Hasselbring, 2025).

En definitiva, estas arquitecturas son una estrategia que se posiciona como un pilar para la transformación digital sostenible al permitir una expansión ágil de las capacidades del software.

1.3.3. API REST

Las Representational State Transfer (APIs REST) constituyen el estándar para la comunicación en arquitecturas modernas de software, estas interfaces utilizan protocolos HTTP estándar (GET, POST, PUT, DELETE) para manipular recursos de manera predecible y escalable Marculescu et al., (2022). El aumento en la variedad y cantidad de aplicaciones que hacen uso de API RESTful ha generado que cada vez se destinen más esfuerzos y recursos tanto al desarrollo como a las pruebas de este tipo de sistemas. Si se adoptan y desarrollan miles de aplicaciones web, un diseño estratégico de APIs asegura la interconexión entre estos sistemas diversos, lo que posibilita a las entidades ajustarse a demandas en aumento Leones et al., (2024).

Sin embargo, para que esta integración sea confiable, la seguridad resulta igualmente fundamental. Para proteger la información que circula a través de estas interfaces, se utilizan estándares como OAuth2¹ y JWT², que controlan quién puede acceder al sistema y bajo qué condiciones, evitando así accesos no autorizados y posibles vulnerabilidades (Leones et al., 2024). Por otro lado, mantener un sistema de versiones claro y una documentación técnica actualizada de las APIs permite incorporar mejoras o nuevas funcionalidades sin afectar el funcionamiento de los servicios ya existentes, lo que garantiza que la plataforma pueda crecer y evolucionar de forma ordenada y sostenible en el tiempo (Leones et al., 2024).

1.3.4. Escalabilidad en PostgreSQL

PostgreSQL se presenta como un sistema gestor de bases de datos relacionales (RDBMS) de código abierto, reconocido por su capacidad para soportar aplicaciones de alto rendimiento (Khan et al., 2023). Con respecto a esta herramienta, la postulan que se orienta a la escalabilidad

¹ **OAuth2.** Es un sistema de permisos que deja que las apps usen datos de usuarios sin que tengan que dar sus claves, confiando la tarea a terceros.

² **JWT.** Este es un formato de token eficiente y protegido, comúnmente usado por OAuth2 para compartir datos del usuario y sus autorizaciones entre diferentes plataformas.

vertical y la gestión robusta de datos estructurados, garantizando el cumplimiento estricto de las propiedades ACID: atomicidad, solidez, aislamiento y continuidad.

Dentro de este ecosistema, PostgreSQL se posiciona como uno de los sistemas gestores de código abierto más robustos y ampliamente adoptados, y con más de cuatro décadas de desarrollo activo, es capaz de gestionar clústeres con múltiples terabytes de datos manteniendo altos niveles de confiabilidad (PostgreSQL Global Development Group, 2026).

Por lo cual, para entornos donde la consistencia y la integridad de los datos son requisitos esenciales, como en sistemas académicos, financieros o de gestión institucional, PostgreSQL se mantiene como una opción técnica de primer nivel, dado que su escalabilidad horizontal en bases de datos SQL introduce complejidades significativas para mantener las garantías ACID en entornos distribuidos (Khan et al., 2023).

1.3.5. Metodologías ágiles de desarrollo

Según los métodos ágiles en el desarrollo constituyen un ámbito laboral ligero que favorece que individuos, grupos y entidades produzcan calidad mediante estrategias ajustables para desafíos complicados (Schwaber & Sutherland, 2020). Al analizar la Tabla 1, se evidencia que Scrum enfatiza la gestión de procesos con iteraciones fijas, mientras XP prioriza prácticas técnicas como TDD, Kanban optimiza flujos continuos limitando WIP, y Lean elimina desperdicios para maximizar valor al cliente; estas diferencias permiten selecciones híbridas según el contexto del proyecto. La comparación destaca cómo XP complementa a Scrum en excelencia técnica, Kanban en flexibilidad operativa y Lean en eficiencia general.

Tabla 1*Tabla comparativa de las metodologías ágiles de desarrollo*

Aspecto	Scrum	XP	Kanban	Lean
Pilares/Valores	Transparencia, inspección, adaptación	Comunicación, simplicidad, feedback, coraje, respeto	Visualizar trabajo, limitar WIP, gestionar flujo	Eliminar desperdicio, valor al cliente, flujo continuo
Roles e iteraciones	Product Owner, Scrum Master, Equipo Desarrollo; Sprints fijos (2-4 semanas)	Cliente on-site, Tracker, programadores; Iteraciones cortas (1-2 semanas)	Sin roles fijos, facilitador opcional; Flujo continuo sin iteraciones	Equipo multifuncional; Ciclos variables, foco continuo
Prácticas clave	Daily Scrum, Review, Retrospective	Pair programming, TDD, integración continua	Tablero Kanban, pull system, políticas explícitas	Mapa de valor, just-in-time, kaizen
Artefactos	Product/Sprint Backlog, Incremento	User Stories, código/refactored	Tablero Kanban, métricas de flujo	Mapa de flujo de valor, métricas de desperdicio

Fuente: Elaboración propia.

En Scrum, el equipo transforma el Product Backlog en incrementos durante Sprints, con inspección constante para adaptarse (Schwaber & Sutherland, 2020). XP complementa con énfasis técnico (e.g., TDD), ideal para calidad en código. Kanban optimiza flujo continuo sin sprints, útil para mantenimiento; Lean elimina desperdicios para eficiencia máxima. Fraguío (2023) resalta su rol en pruebas frecuentes para sistemas web ágiles.

Su flexibilidad resulta crucial en la creación de sistemas modernos, como los módulos web para la gestión de proyectos de voluntariado, donde Fraguío (2023) complementa este enfoque señalando que el uso de estas metodologías ágiles permite realizar pruebas de validación frecuentes, donde el propietario del producto proporciona una retroalimentación que elimina la brecha comunicacional entre desarrolladores y usuarios.

1.3.6. Diseño UX/UI: Adopción y satisfacción del usuario

La Experiencia de Usuario (UX) abarca cómo se siente y qué percibe una persona cuando interactúa con una web o sistema digital, incluyendo sus emociones, impresiones y reacciones durante toda la interacción. Se ratifica que el diseño tiene como propósito fundamental optimizar la interacción global del usuario. Al analizar sus requerimientos, objetivos y dificultades, busca desarrollar soluciones que resulten accesibles, prácticas y de calidad para el consumidor final (Stevens, 2024).

Como rama de la interacción humano-computadora (HCI), la UX se contextualiza con la lógica de negocio para diseñar flujos de navegación adaptados al comportamiento real de los usuarios (Gordon Graell, 2023).

1.3.7. Diferenciación técnica entre UX y UI

Es fundamental distinguir la UX de la Interfaz de Usuario (UI). Mientras que la UX se enfoca en la estructura de la información y la fluidez del sentido, la UI se centra en los elementos visuales y puntos de contacto, definiendo qué observa el usuario y cómo interactúa con elementos gráficos (colores, fuentes, botones). En conjunto, ambas disciplinas aseguran que cada prototipo web logre la característica intuitiva, visualmente coherente y funcionalmente lógica (Stevens, 2024).

1.3.8. Gestión de proyectos y eficiencia organizacional (Soluciones informáticas existentes para la gestión de proyectos)

La automatización y centralización de procesos administrativos resulta crucial para la eficiencia institucional. Ramesh et al., (2025) presentan en su estudio “Design and Implementation of Project and Finance Management System Using PHP and MySQL” que el uso de módulos de gestión asegura que los proyectos se cumplan según lo planificado, reduciendo significativamente la carga operativa en los departamentos administrativos. Esta lógica ofrece una plataforma para la administración de registros exactos, permitiendo la toma de decisiones informadas mediante reportes y análisis de desempeño en tiempo real.

Siguiendo la metodología de evaluación comparativa de Jihan y Mahummad, (2024) quienes analizan las herramientas bajo el estándar del Project Management Body of Knowledge (PMBOK), es posible determinar la idoneidad de las soluciones más representativas del mercado. En la Tabla 2 se presenta este análisis, incorporando el modelo de licenciamiento y costos, un factor que define la viabilidad de adopción a largo plazo.

Tabla 2*Tabla comparativa de las herramientas de gestión de proyectos*

Herramienta	Escala de proyecto	de	Soporte PMBOK	Modelo de costos y licenciamiento	Limitaciones técnicas y económicas
Trello	Pequeña Sencilla	/	Bajo: Sin áreas avanzadas.	Open Source/ Pago por usuario para automatización avanzada.	Inadecuado para macro-proyectos; funciones limitadas en versiones sin costo. Restricción de responsable único por tarea; las funciones avanzadas requieren planes de pago más costosos.
Asana	Grande Compleja	/	Alto: Módulos diversos y reportes.	Open Source/ Planes de pago escalonados por volumen de usuarios.	
Monday.com	Corporativa / Diversa		Medio-Alto: Flujos de negocio.	Suscripción recurrente (SaaS) condicionada a paquetes mínimos de usuarios.	Alta carga económica recurrente en despliegues institucionales masivos.
Microsoft Project	Macro-proyectos		Muy Alto: Costos y cronogramas.	Licenciamiento corporativo cerrado o (adquisición suscripción premium).	Elevado costo de adquisición; dependencia financiera del ecosistema propietario.
Jira Software	Ingeniería / Scrum	/	Muy Alto: Calidad e integración.	Suscripción escalonada (SaaS) o licencias por infraestructura de servidor.	Incremento significativo de costos de licencia al escalar la cantidad de usuarios administrativos.

Fuente: Elaboración propia.

Este análisis evidencia que la efectividad de una plataforma no es intrínseca, sino que depende de su alineación con las condiciones específicas y el presupuesto del proyecto (Jihan & Mahummad, 2024). Por tanto, la adopción tecnológica en entornos institucionales debe priorizar factores como la soberanía de los datos y evaluar rigurosamente el modelo de negocio de estas plataformas. En muchas soluciones comerciales, la cobertura de áreas críticas de gestión se encuentra restringida por barreras económicas, obligando a las organizaciones a asumir altos costos recurrentes por usuario para acceder a funcionalidades avanzadas.

1.3.9. Seguridad en aplicaciones web

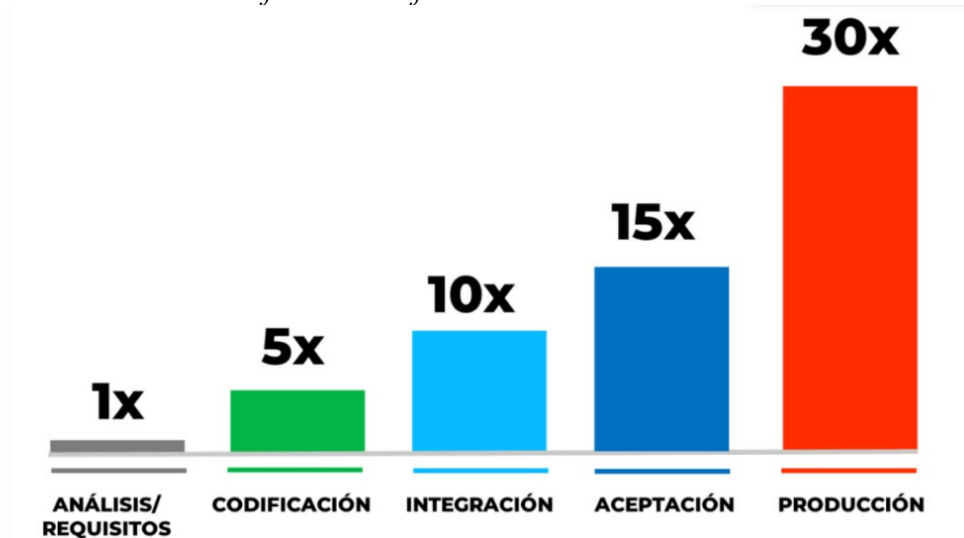
La seguridad en aplicaciones web es una necesidad crítica que debe abordarse desde las etapas iniciales del desarrollo de software, se aluden a una carencia significativa de personal especializado en seguridad dentro de las instituciones educativas, lo que traslada la responsabilidad técnica directamente al desarrollador web para garantizar la protección del

sistema (Al-Ibrahim & Shams, 2014). Tradicionalmente, las empresas de desarrollo han enfocado el diseño y la implementación del software únicamente a la funcionalidad, sin incorporar prácticas de seguridad durante las fases de construcción, esperando hasta la etapa de pruebas para identificar y solventar vulnerabilidades, es decir, trabajando bajo un modelo reactivo (Sancho-Nuñez, 2021).

Sin embargo, retrasar la solución de fallos de seguridad hasta fases tardías incrementa exponencialmente el coste del desarrollo y puede ocasionar rediseños y cambios excesivos en la implementación. Al respecto, el Instituto Nacional de Estándares y Tecnología (NIST, citado en Sancho Núñez, 2021) cuantifica que solucionar una vulnerabilidad en la fase de post-producción resulta hasta 30 veces más costoso que hacerlo durante el diseño, la toma de requisitos o la definición de la arquitectura, tal como se refleja en la Figura 2. La curva o barras muestran que el coste es mucho menor si los fallos se detectan en fases tempranas (requisitos, diseño) y aumenta exponencialmente si se detectan en fases finales (pruebas, post-producción).

Figura 2

Coste de resolución de fallos de la fase del ciclo de vida donde se detecta.



Nota. Tomado de "Implementación de un modelo de desarrollo de software seguro", por J. Núñez, S. 2021, Universidad de Extremadura. Fuente original: NIST. Derechos reservados.

Ante este escenario, la necesidad de reducir el riesgo de exposición y los costes adicionales generados por vulnerabilidades tardías ha llevado a incorporar procesos de seguridad desde las primeras fases del ciclo de vida del software. Este enfoque preventivo busca que la seguridad no afecte negativamente a los tiempos de implementación, sino que, por el contrario, permita construir sistemas más robustos que los desarrollados bajo esquemas reactivos, evitando así

posibles pérdidas de confianza por parte de los usuarios y las consecuentes afectaciones a la continuidad del servicio (Sancho-Núñez, 2021).

1.3.10. Frameworks Frontend

Vue.js. La documentación oficial de ratifica que este framework se especializa en la creación de interfaces de usuario basadas en componentes reactivos (Vue.js, (s.f.)). Su estructura modular facilita el manejo de la interfaz mediante tecnologías estándar como HTML, JavaScript y CSS. Con el apoyo de Node.js, es posible renderizar componentes de forma eficiente, permitiendo una interacción humano-computadora personalizada y escalable.

React. Se describen como una biblioteca de JavaScript para construir interfaces de usuario declarativas mediante un modelo de componentes y Virtual DOM, destacando que su arquitectura basada en componentes reutilizables permite mejorar el rendimiento y la creación de interfaces dinámicas (Keshari et al., 2023). Su enfoque en la reutilización de componentes y el estado gestionado con hooks permite interfaces dinámicas y de alto rendimiento en aplicaciones escalables (White et al., 2024)

Angular. Se considera como una plataforma empresarial de desarrollo web basada en TypeScript que incorpora inyección de dependencias, módulos estructurados y herramientas integradas, señalando que su ecosistema facilita el desarrollo de aplicaciones robustas y escalables en entornos de nube y móvil (Nadipalli, 2024). Su arquitectura basada en módulos y directivas facilita el desarrollo de aplicaciones empresariales robustas y esta diseñado con una base de actualización muy sencilla (Gongalves, 2021)

1.3.11. Frameworks Backend

Nest.js. Pham (2020) analiza a NestJS como el marco de desarrollo que consolida la lógica de negocio en sistemas web modernos. Al ejecutarse sobre Node.js e integrar TypeScript como lenguaje principal, permite aplicar principios de programación orientada a objetos bajo una arquitectura de módulos, controladores y servicios. Así, los errores relacionados con el tipado de datos se detectan en tiempo de compilación, lo que reduce los problemas en tiempo de ejecución y mejora la calidad del código (Lando & Hasselbring, 2025).

Como explica el propio creador del framework, Myśliwiec (citado por Marín, 2025) el problema con herramientas muy flexibles como Express es que, al manejar proyectos grandes, el código suele volverse un caos. Para solucionar esto, NestJS se lleva la arquitectura de "monolito modular". Esto significa que obliga a agrupar funcionalidades concretas mediante

decoradores `@Module` (por ejemplo, aislando un módulo solo para los voluntarios y otro para los proyectos). Al forzar estos límites desde el inicio, se logra que el sistema se mantenga ordenado y fácil de escalar a largo plazo (MySliwiec, 2017).

Spring Boot.: define Spring Boot como un framework que simplifica la creación de aplicaciones Java standalone mediante configuración automática y servidores embebidos, señalando que su arquitectura orientada a microservicios y el uso de anotaciones facilitan el desarrollo de sistemas empresariales escalables con estrategias de despliegue en contenedores (Ahmad, 2024). Su enfoque en microservicios y anotaciones facilita el desarrollo de sistemas empresariales escalables con soporte nativo para contenedores (Pivotal, 2024).

CodeIgniter. Se presenta como un framework PHP ligero y rápido que sigue el patrón MVC para desarrollar aplicaciones web dinámicas. Su estructura simple con bibliotecas mínimas y carga dinámica acelera el desarrollo de proyectos medianos sin complejidades innecesarias (Ellislab/CodeIgniter, 2022).

1.3.12. Lenguaje de modelado unificado (uml)

En la ingeniería de software UML constituye el marco estructural para la especificación y visualización de flujos de procesos y del diseño de un sistema de software Kumarapeli et al. (2007) citados en (Modi et al., 2021) indican que el uso de diagramas como por ejemplo los de casos de uso y actividades permiten que los interesados en el software comprendan el funcionamiento del sistema sin necesidad de profundizar en el código fuente. Este modelado facilita la comunicación técnica y asegura que todas las aplicaciones web estén documentadas durante el desarrollo de las mismas.

Tabla 3

Tipos de diagramas Existentes del Lenguaje de Modelado Unificado (UML)

Tipos de diagramas UML	Tipos de diagramas UML
Diagrama de casos de uso	Diagrama de comunicación / colaboración
Diagrama de procesos	Diagrama de componentes
Diagrama de actividades	Diagrama de despliegue

Nota. Tomado de "UML Diagrams in Software Engineering Research: A Systematic Literature Review", por H. Koç et al., 2021, *MDPI Proceedings*, 74(1), 13. CC BY 4.0 (<https://doi.org/10.3390/proceedings2021074013>).

La Tabla 3 indicada como Koç et al. (2021) sintetiza los tipos de diagramas UML agrupados en sus dos categorías principales, tal como se muestra.

1.3.13. Pruebas de software

La etapa en la que un software es puesto a verificación o aprobación es indispensable en cualquier ciclo de codificación. Su propósito principal va más allá de una simple comprobación rutinaria; se trata de verificar y validar el sistema para garantizar que cumple con los requisitos para los que fue diseñado, operando con la menor cantidad de errores posibles (Pantaleo & Rianaudo, 2015).

Para comprender mejor cómo se organiza, se propone clasificar estas verificaciones en dos grandes categorías: las funcionales y las no funcionales. Las funcionales se enfocan en el comportamiento externo del producto construido. Es decir, comprueban que la aplicación cumpla con las tareas asignadas, sin detenerse a analizar cómo está construido el código por dentro (Dreiser, 2024). Dentro de este grupo destaca la técnica de caja negra (*black box testing*), donde quien evalúa interactúa con el sistema como lo haría un usuario final: ingresa información y verifica que la respuesta obtenida sea la esperada, ignorando la lógica interna (Pirdaus & Hidayana, 2024). En contraste, cuando es necesario examinar la arquitectura subyacente y el código fuente, se aplican las pruebas de caja blanca (*white box testing*). Dreiser, (2024) explica que este enfoque analiza la estructura interna, revisando a fondo las rutas lógicas y los flujos de ejecución, lo cual es clave para detectar vulnerabilidades de seguridad y optimizar la eficiencia de la programación.

Ahora bien, un sistema no solo debe cumplir sus funciones, sino que debe hacerlo en óptimas condiciones. Es ahí donde entran las consideradas no funcionales, encargadas de testear atributos que trascienden la operatividad básica, como el desempeño, la facilidad de utilización, la compatibilidad y la transferibilidad. Definitivamente estos factores son los pilares que definen la verdadera calidad del software a nivel internacional, siendo la rapidez de respuesta, la confiabilidad y la seguridad los aspectos más críticos cuando el proyecto ya se encuentra en un entorno de producción.

Dentro de las pruebas no funcionales, las pruebas denominadas como de usabilidad ocupan un lugar especialmente relevante, aquí se evalúa la perfección de las funcionalidades del producto. Su objetivo no es únicamente detectar errores técnicos, sino medir qué tan fácil, eficiente e intuitivo resulta el uso del software para una persona real. De acuerdo con la norma ISO 9241-11:2018, la usabilidad se define a partir de tres dimensiones: eficacia, eficiencia y estabilidad. Para llevar a cabo estas evaluaciones, la literatura especializada reconoce dos grandes enfoques. Por un lado, los métodos empíricos, que implican la observación directa de usuarios reales completando tareas predefinidas; entre sus técnicas más utilizadas se encuentran el

protocolo de pensamiento en voz alta (*think-aloud*), el *cognitive walkthrough* y el cuestionario SUS (*System Usability Scale*), que permite obtener una puntuación cuantitativa del nivel de usabilidad percibida. Por otro lado, los métodos de inspección, realizados por expertos sin necesidad de usuarios reales, entre los que sobresale la evaluación heurística, basada en principios establecidos como los diez heurísticos de Nielsen. Además, métodos adicionales dentro de esta categoría, como el recorrido heurístico, la inspección de consistencia y la inspección basada en personas (*persona-based inspection*), lo que evidencia la riqueza metodológica disponible para el evaluador (Ntoa, 2024).

Por su parte, la correcta ejecución de cualquier tipo de prueba requiere de una planificación previa estructurada, lo que en ingeniería de software se conoce como plan de pruebas (test plan). Según el estándar IEEE 829, las pruebas definen el alcance, el enfoque, y los recursos de un proyecto (IEEE SA, 2005). En ellas se especifican los criterios de verificación, los casos de prueba a ejecutar y las responsabilidades del equipo. Así mismo, señalan que contar con un plan de pruebas bien definido no solo organiza el proceso, sino que también garantiza la trazabilidad entre los requisitos del sistema y las verificaciones realizadas, lo cual es fundamental para demostrar la calidad del producto (Pressman & Maxim, 2014).

En definitiva, estos enfoques de evaluación son complementarios e interdependientes para lograr un producto de calidad. Mientras que las pruebas funcionales aseguran que el sistema hace lo que tiene que hacer, las no funcionales garantizan que lo hace con la solidez y eficiencia que demanda la lógica del negocio.

CAPÍTULO II. MATERIALES Y MÉTODOS

En este capítulo se abordan las pautas metodológicas que guían el desarrollo de la presente investigación, junto con los procedimientos y técnicas utilizados para alcanzar los objetivos propuestos. De igual manera, se expone de forma estructurada el proceso general empleado en la construcción del sistema, detallando los enfoques adoptados durante las fases del proyecto para dar solución a la problemática planteada.

2.1. Aspectos generales de la investigación

El presente trabajo de titulación se desarrolló bajo una investigación de tipo aplicada, debido a que tiene como propósito diseñar y desarrollar una solución tecnológica que responda a una problemática concreta identificada en la gestión de proyectos de voluntariado. La propuesta plantea optimizar procesos relacionados con la administración de proyectos, registro de participantes voluntarios, asignación de líderes, control de participantes, seguimiento del estado de los proyectos y generación de información para la toma de decisiones.

La investigación adoptó un enfoque cualitativo, ya que se fundamenta en la interpretación de las necesidades, opiniones y experiencias de los actores involucrados en el proceso. Este enfoque permitió comprender la situación actual, identificar las principales dificultades y establecer los requerimientos necesarios para el desarrollo del sistema web propuesto.

Para la recolección de información se realizó una observación general del proceso actual, la cual permitió evidenciar que no existe un sistema automatizado para la gestión de los proyectos de voluntariado. Esta observación fue utilizada como técnica complementaria, con el fin de confirmar que la información y las actividades se manejan de forma manual o dispersa, lo que genera la necesidad de una herramienta tecnológica que facilite el control y seguimiento del proceso.

Adicionalmente se aplicaron entrevistas semiestructuradas y un grupo focal cómo se presenta en el anexo1. Las entrevistas se aplicaron mediante un guion previamente elaborado, con preguntas orientadas a conocer el funcionamiento actual del proceso, los roles participantes, las dificultades existentes y las funcionalidades requeridas para la plataforma. Por su parte, el grupo focal permitió reunir diferentes opiniones de los usuarios involucrados, contrastar necesidades comunes y validar aspectos importantes para el diseño del sistema.

En conjunto, las técnicas aplicadas permitieron obtener información relevante para definir los requerimientos funcionales y no funcionales del sistema, así como estructurar las historias de usuario y las funcionalidades principales de la solución propuesta.

2.1.1. Propósito

El propósito del presente trabajo fue desarrollar una solución web para mejorar la gestión de los proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra, mediante la centralización de la información y la organización de los procesos relacionados con proyectos, líderes, voluntarios y áreas de voluntariado.

La propuesta respondió a las limitaciones del proceso manual que se utilizaba en el área de voluntariado, facilitando el registro, seguimiento y consulta de la información. Asimismo, el sistema permitió disponer de reportes estadísticos orientados al control de los proyectos y al apoyo en la toma de decisiones dentro del área responsable.

2.1.2. Alcance

El proyecto llegó hasta el desarrollo funcional de un sistema web para la gestión interna de los proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra. El sistema contempló la administración de proyectos de voluntariado, la gestión de usuarios según su rol, el registro de solicitudes de participación, el seguimiento de los participantes y la generación de reportes estadísticos.

En el módulo de administración del voluntariado, el coordinador pudo registrar, consultar, actualizar y eliminar proyectos. Durante la creación de cada proyecto, el sistema permitió seleccionar el líder responsable, escoger el área o categoría de voluntariado correspondiente y registrar información descriptiva del proyecto, incluyendo datos informativos como los beneficiarios.

De la misma manera, el sistema incluyó funcionalidades para que los usuarios consultaran los proyectos a los que han sido asignados como responsables y solicitaran su planificación mediante el registro de información de participantes de determinada carrera y semestre. Esta información permitió alinear los indicadores relacionados con la participación en los proyectos de voluntariado.

A su vez, el líder de proyecto pudo visualizar los proyectos asignados, actualizar el estado del proyecto según su avance y eliminar a participantes que ya no formaron parte del voluntariado, manteniendo actualizada la información de participación.

Para cerrar, el sistema incorporó un módulo de seguimiento y reportes, mediante el cual el coordinador pudo visualizar información consolidada sobre proyectos, participantes e indicadores del voluntariado. El alcance del trabajo se limitó al desarrollo funcional del sistema web y a la validación de sus principales funcionalidades en un entorno controlado, sin contemplar la integración con sistemas externos ni desarrollo de una aplicación móvil.

2.2. Metodología de la propuesta tecnológica

La metodología ágil Scrum fue elegida para desarrollar el sistema web. La elección de la metodología Scrum se basó en su característica para organizar el desarrollo del software mediante ciclos cortos de trabajo denominados *Sprints*, lo que propició una comunicación continua entre los actores del proceso a través de reuniones periódicas de seguimiento. Esta forma de trabajo favoreció la implementación gradual de ajustes, al revisar al final de cada Sprint los incrementos entregados y constatar que el flujo del proyecto se dirigía en la dirección correcta. De igual manera, esta perspectiva promovió la priorización de los requerimientos mediante una lista ordenada de historias de usuario (Product Backlog), garantizando así una ejecución optimizada de los requerimientos y evitando problemas que pudieran surgir por agregar funcionalidades no contempladas en el plan inicial. Además, su carácter iterativo permitió la posibilidad de verificar la lógica y el comportamiento del sistema durante todo el proceso de desarrollo, en vez de retrasar las pruebas hasta la última etapa.

Con el propósito de ilustrar el funcionamiento de la metodología seleccionada, se presenta a continuación el marco de trabajo Scrum, en el que se evidencian los roles, eventos y artefactos que estructuran el desarrollo iterativo del sistema, tal como se observa en la

Figura 3.

Figura 3
Marco de trabajo Scrum



Nota. Tomado de “*What is Scrum? [Definition + How It Works]*”, por R. Patel, 2025, Space-O Technologies. <https://www.spaceo.ca/glossary/software-terms/what-is-scrum/>

2.2.1. Planificación del Product Backlog

La fase de planificación estableció las funcionalidades principales del sistema web a partir de las necesidades identificadas en el proceso actual de gestión de los proyectos de voluntariado. Para ello, se definieron historias de usuario que permitieron describir los requerimientos desde la perspectiva de los actores que interactuaron con el sistema: coordinador, líder de proyecto y usuario voluntario.

Las historias de usuario permitieron representar de manera clara las acciones esperadas por cada tipo de usuario dentro del sistema. Estas se formularon considerando el rol del usuario, la funcionalidad requerida y el beneficio esperado dentro del proceso de gestión del voluntariado. De esta forma, en la Tabla 4 se estableció una base inicial para organizar el desarrollo de los módulos del software.

Para garantizar una correcta trazabilidad técnica y facilitar la posterior etapa de codificación, la tabla se estructuró detallando parámetros clave propios de las metodologías ágiles. Cada historia incluye su identificador (ID) y nombre, seguidos del módulo al que pertenece y el rol responsable de ejecutar la acción. Asimismo, se incorporaron estimaciones de esfuerzo (h) e importancia para priorizar el cronograma de desarrollo. Por último, se documentó la descripción narrativa de cada requerimiento, sus respectivos criterios de aceptación (las condiciones obligatorias que el código debe cumplir para dar por válida la funcionalidad) y las dependencias lógicas entre tareas, asegurando un flujo de trabajo ordenado y coherente con los objetivos del proyecto.

El análisis detallado de estas variables operativas permitió estructurar el alcance del sistema de manera organizada. Este proceso resultó vital para mitigar riesgos técnicos, evitar desviaciones arquitectónicas durante el desarrollo y asegurar que cada requerimiento respondiera a un objetivo previamente validado.

Consecuentemente, la Tabla 4 se consolidó como el instrumento central de organización metodológica. Su formato matricial desglosó el esfuerzo de programación, priorizó las tareas según su nivel de criticidad técnica y estableció las dependencias lógicas entre cada componente. De este modo, la matriz guió la implementación secuencial del código a lo largo de los Sprints de desarrollo del software, garantizando que todos los criterios de aceptación se cumplieran rigurosamente antes de aprobar la integración de cada nuevo módulo.

Tabla 4*Historias de usuario del sistema web de voluntariado*

ID	Nombre	Módulo	Rol	Esfuerzo (h)	Importancia	Descripción de historia de usuario	Criterios de aceptación	Dependencias
1	Autenticación y control de roles	Transversal	Coordinador, Líder o Voluntario	16	120	Como actor del proceso con rol de coordinador, líder o voluntario, requiero autenticarme en la plataforma mediante credenciales de acceso, con el fin de ingresar al entorno de trabajo correspondiente a mis permisos.	La plataforma debe redirigir al panel específico del perfil tras una validación exitosa. Se emitirá un mensaje de error visible cuando las credenciales no coincidan con los registros.	-
2	Creación y asignación de proyecto de voluntariado	Administración de voluntariado	Coordinador	32	200	Como coordinador, requiero crear proyectos de voluntariado y asignarlos a un líder/docente responsable, para iniciar formalmente el ciclo del proyecto en el sistema.	El formulario debe validar todos los campos requeridos del proyecto. Debe incluir un selector para asignar un líder registrado. El proyecto debe quedar visible en el panel del líder asignado y en el panel general del coordinador.	1
3	Registro de participantes en el proyecto	Gestión del proyecto asignado	Líder/Docente	20	200	Como líder/docente, necesito registrar a los voluntarios (estudiantes) que participarán en el proyecto	El líder solo podrá registrar participantes en proyectos asignados a su perfil. Se deben capturar su rol o si es	1, 2

						asignado, ingresando sus datos académicos, para formalizar su participación. Como líder/docente, necesito registrar las actividades a realizarse dentro del proyecto, para organizar y dejar constancia del plan de trabajo antes de solicitar la ejecución. Como líder/docente, requiero solicitar formalmente la ejecución del proyecto una vez completada la planificación, para que el coordinador pueda revisar y autorizar el cambio de estado. Como coordinador, requiero revisar las solicitudes	externo el dato informativo del voluntario. El voluntario registrado debe aparecer en el listado. El sistema debe permitir ingresar y editar las actividades planificadas. Las actividades deben quedar asociadas al proyecto correspondiente. El líder debe poder modificarlas antes de enviar la solicitud de ejecución. Debe existir una acción para enviar la solicitud de ejecución al coordinador. La solicitud debe quedar visible y pendiente en el panel del coordinador. Si el coordinador indica cambios necesarios, el líder debe poder corregir y re-enviar. El coordinador debe ver las solicitudes de proyectos	
4	Planificación de actividades del proyecto	Gestión del proyecto asignado	Líder/Docente	20	180			1, 2, 3
5	Solicitud de ejecución del proyecto	Gestión del proyecto asignado	Líder/Docente	16	180			1, 2, 4
6	Autorización del cambio de estado por el	Administración de voluntariado	Coordinador	16	180			1, 2, 5

	coordinador					de cambio de estado enviadas por el líder y autorizar o solicitar correcciones, para controlar el avance formal del ciclo del proyecto.	pendientes de autorización. Debe poder aprobar el cambio de estado o devolver la solicitud indicando cambios necesarios. La decisión debe reflejarse de inmediato en el estado del proyecto.
7	Registro de actividades realizadas y evidencias	Gestión del proyecto asignado	Líder/Docente	20	160	Como líder/docente, necesito registrar las actividades ejecutadas durante el proyecto junto con fotografías y registros de salidas, para documentar el desarrollo real del voluntariado.	El sistema debe permitir registrar evidencias con descripción y fecha. Debe permitir adjuntar registros y quedar vinculadas al proyecto y visibles para el coordinador. El líder debe completar el informe de ejecución antes de poder cambiar el estado a finalizado.
8	Cierre del proyecto e informe de ejecución	Gestión del proyecto asignado	Líder/Docente	16	160	Como líder/docente, necesito registrar el informe de ejecución y cambiar el estado del proyecto a finalizado, para cerrar formalmente el ciclo del proyecto en el sistema.	Al confirmar el cierre, el estado del proyecto debe actualizarse y quedar registrado para el coordinador.

9	Visualización de información centralizada	Reportes de voluntariado	Coordinador	32	160	Como coordinador, quiero acceder a un panel con información consolidada de todos los proyectos, participantes e indicadores del voluntariado, para tener una visión global y tomar decisiones informadas.	El panel debe mostrar el estado actual de todos los proyectos registrados. Debe permitir filtrar información por carrera, semestre, facultad u otros indicadores. Debe incluir los datos de participantes registrados por los líderes y las evidencias de los proyectos finalizados.	1, 2, 3, 8
---	---	--------------------------	-------------	----	-----	---	--	------------

Fuente: Elaboración propia.

Posteriormente, se realizó el análisis de las historias de usuario con el propósito de relacionar cada requerimiento con el módulo correspondiente del sistema. Este análisis permitió identificar las funcionalidades necesarias, los actores involucrados y la prioridad de desarrollo de cada historia, tal y como se visualiza en la Tabla 5. Además, facilitó la organización de los requerimientos de acuerdo con la lógica funcional del sistema web.

Finalmente, se elaboró la planificación de las historias de usuario, en la cual se organizó el desarrollo de las funcionalidades de acuerdo con su prioridad y dependencia funcional. Esta planificación y análisis de historial de usuario permitió definir el orden en que se construyeron los módulos del sistema, considerando primero las funcionalidades base, como la gestión de usuarios, roles y proyectos, y posteriormente las funciones relacionadas con solicitudes de participación, seguimiento y reportes. Tabla 5.

Tabla 5*Tabla de análisis de las historias de usuario*

Prioridad	N.º historia	ID tarea	Tareas	Horas esfuerzo
200	2	T2-1	Diseñar la estructura de base de datos para proyectos de voluntariado.	4
		T2-2	Diseñar la interfaz del formulario de creación de proyectos.	6
		T2-3	Implementar la funcionalidad de creación de proyectos.	6
		T2-4	Implementar el selector para asignar líder registrado al proyecto.	4
		T2-5	Implementar visualización del proyecto en panel del líder y coordinador.	6
		T2-6	Realizar pruebas de creación, asignación y visualización del proyecto.	6
200	3	T3-1	Diseñar la interfaz de registro de participantes para el líder.	4
		T3-2	Implementar campos de carrera, semestre y facultad del voluntario.	4
		T3-3	Validar que el líder solo registre participantes en sus proyectos asignados.	2
		T3-4	Implementar asociación del voluntario al proyecto correspondiente.	6
		T3-5	Bloquear la auto-inscripción del voluntario en el sistema.	2
		T3-6	Realizar pruebas de registro, asociación y restricción.	2
		T5-1	Diseñar la interfaz de solicitud de ejecución del proyecto.	2
		T5-2	Implementar la acción de envío de solicitud de ejecución al coordinador.	4
180	5	T5-3	Mostrar la solicitud como pendiente en el panel del coordinador.	4
		T5-4	Permitir al líder corregir la planificación y re-enviar la solicitud.	4
		T5-5	Realizar pruebas de solicitud, devolución y re-envío.	2
		T6-1	Diseñar panel del coordinador con solicitudes pendientes de autorización.	2
		T6-2	Implementar la acción de aprobación del cambio de estado.	4
180	6	T6-3	Implementar la acción de devolución con indicación de cambios necesarios.	4
		T6-4	Reflejar la decisión del coordinador en el estado visible del proyecto.	4
		T6-5	Realizar pruebas de aprobación y devolución de solicitudes.	2
		T4-1	Diseñar la interfaz de registro de actividades planificadas.	4
		T4-2	Implementar el ingreso y edición de actividades a realizarse.	6
180	4	T4-3	Asociar las actividades planificadas al proyecto correspondiente.	4
		T4-4	Realizar pruebas de registro y edición de actividades planificadas.	6
160	7	T7-1	Diseñar la interfaz de registro de actividades realizadas y evidencias.	2

160	8	T7-2	Implementar registro de actividades ejecutadas con descripción y fecha.	6
		T7-3	Implementar carga de fotografías y registros de salidas de campo.	6
		T7-4	Vincular evidencias al proyecto para visualización del coordinador.	4
		T7-5	Realizar pruebas de registro y visualización de evidencias.	2
		T8-1	Diseñar la interfaz de registro del informe de ejecución.	4
		T8-2	Implementar el ingreso del informe de ejecución por parte del líder.	6
		T8-3	Implementar la acción de cambio de estado ha finalizado tras completar el informe.	4
		T8-4	Realizar pruebas de registro del informe y cierre del proyecto.	2
160	9	T9-1	Diseñar la estructura de consulta centralizada de proyectos y participantes.	6
		T9-2	Diseñar la interfaz del panel de reportes consolidado.	6
		T9-3	Implementar filtros por carrera, semestre, facultad e indicadores.	8
		T9-4	Integrar evidencias de proyectos finalizados en el panel.	4
		T9-5	Realizar pruebas de filtrado, visualización e integridad de datos.	8
120	1	T1-1	Diseñar la interfaz de autenticación con usuario y contraseña.	4
		T1-2	Implementar la validación de credenciales de acceso.	4
		T1-3	Configurar la redirección del usuario según su rol.	4
		T1-4	Implementar mensajes de error por credenciales incorrectas.	2
		T1-5	Realizar pruebas de autenticación y control de roles.	2

Fuente: Elaboración propia.

Una vez definido el Product Backlog y estimado el esfuerzo requerido para cada historia de usuario, se estructuró el desarrollo del sistema en cinco Sprints, cada uno con una duración de dos semanas.

Para la planificación se consideró una jornada de trabajo de cinco días a la semana, con un promedio de seis horas diarias, lo que representa una capacidad aproximada de 60 horas por Sprint. Esta planificación permitió equilibrar la carga de trabajo y asegurar entregables funcionales al finalizar cada iteración.

La Tabla 6 presenta la distribución de historias de usuario por Sprint, considerando las horas de esfuerzo estimadas y las dependencias funcionales entre ellas.

Tabla 6*Distribución de historias de usuario en Sprints*

Sprint	Historias de usuario incluidas	Horas esfuerzo	de	Duración (semanas)
Sprint 1	H1. Autenticación y control de roles; H2. Creación y asignación de proyecto de voluntariado	48		2
Sprint 2	H3. Registro de participantes en el proyecto; H4. Planificación de actividades del proyecto	40		2
Sprint 3	H5. Solicitud de ejecución del proyecto; H6. Autorización del cambio de estado por el coordinador	32		2
Sprint 4	H7. Registro de actividades realizadas y evidencias; H8. Cierre del proyecto e informe de ejecución	36		2
Sprint 5	H9. Visualización de información centralizada	32		2
Total	—	188		10

Fuente: Elaboración propia.

2.2.2. Fase de diseño

Durante la fase de diseño se analizó la información obtenida en la etapa de levantamiento de requisitos, considerando las necesidades de la Dirección de Identidad y Misión de la PUCE Ibarra en el área de voluntariado. A partir de este análisis, se definió la estructura funcional del sistema web para la gestión de proyectos de voluntariado, cuyo propósito es centralizar la información, registrar la participación de docentes y estudiantes, gestionar proyectos asociados a distintas categorías de voluntariado y generar reportes estadísticos de forma automática.

Para representar el funcionamiento del sistema se elaboraron los diagramas UML necesarios, los cuales permitieron visualizar la interacción entre los actores y las funcionalidades principales del sistema. Entre estos diagramas se incluyeron el diagrama de casos de uso, donde se identifican los roles que intervienen en el proceso, tales como los responsables de la gestión de voluntariado, docentes y estudiantes voluntarios. Estos actores interactúan con funcionalidades relacionadas con la administración de proyectos, registro de participantes, consulta de información y generación de reportes estadísticos.

Asimismo, los diagramas de diseño contribuyeron a definir la arquitectura de la aplicación, la organización de los módulos del sistema y el diseño de la base de datos. Esto permite establecer una solución coherente con la problemática identificada, especialmente en cuanto a la falta de

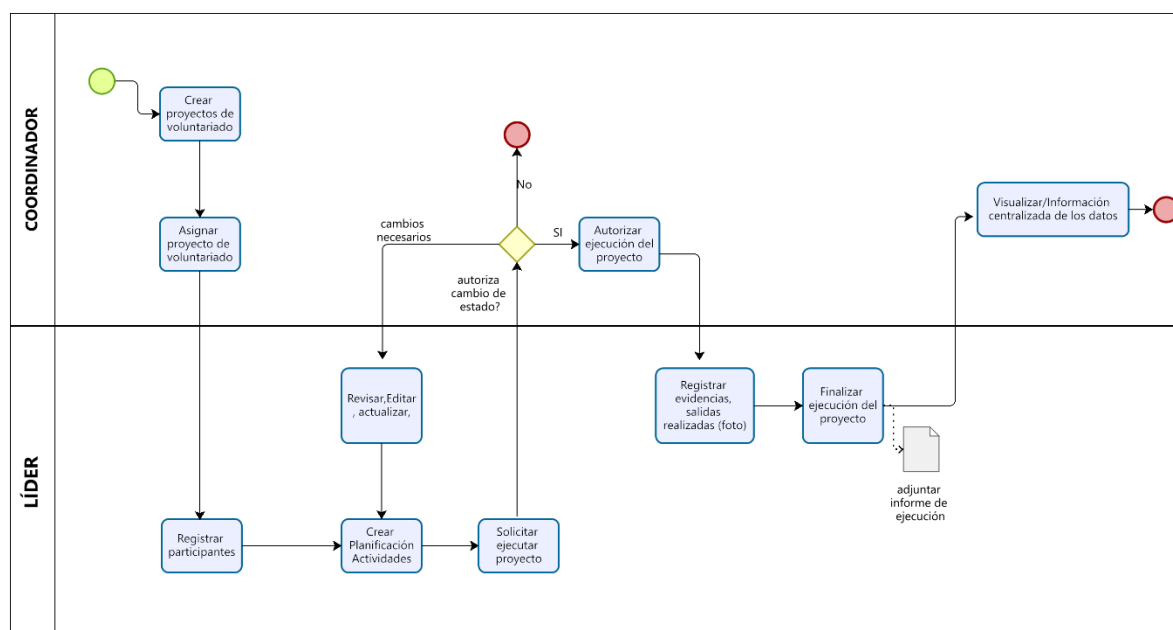
centralización de la información, la ausencia de trazabilidad de los participantes y la necesidad de contar con indicadores oportunos para la toma de decisiones y procesos de evaluación institucional.

El sistema se desarrolló con tecnologías software que permitieron la escalabilidad y buen rendimiento. Además, se emplearon principios de diseño de experiencia de usuario, así como el diseño de interfaces. Una base de datos PostgreSQL que manejó la lógica del negocio. Todo se asentará en una arquitectura modular para mejorar la interacción de los tres tipos de usuario: Coordinador, Líder, voluntario. Así mismo el manejo con los módulos del sistema web de voluntariado:

El desarrollo contó con una secuencia bien estructurada, iniciando con el diseño de la aplicación y la elaboración de un diagrama de casos de uso y un diagrama de flujo, los cuales detallarán la interacción entre los componentes y los actores. A continuación, se presenta la sección de diagramas.

El diagrama de proceso del sistema de Voluntariado explica cómo es la interacción del coordinador, líder (docente) y usuario común (estudiante), como se observa en la Figura 4.

Figura 4 *Diagrama de procesos del sistema de voluntariado*

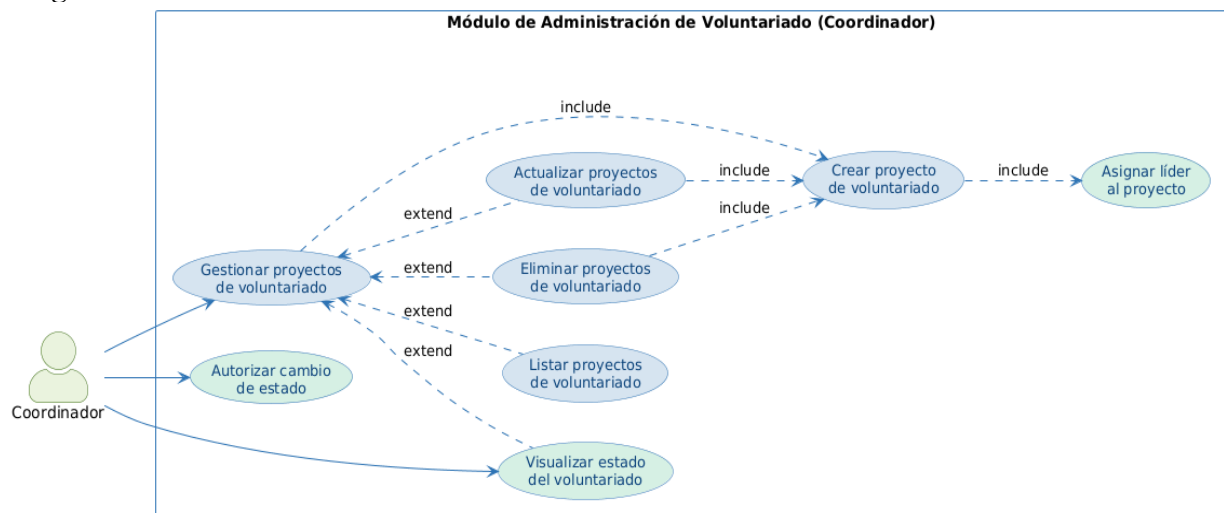


Fuente: Elaboración propia.

Módulo administración del voluntariado. En esta sección, el coordinador administra las áreas de voluntariado y los beneficiarios asociados a los proyectos, quienes representan los lugares o entidades donde se ejecutan las acciones de voluntariado. Para las áreas, el sistema

permite crear, editar y cambiar su estado (activa o inactiva), mientras que para los beneficiarios permite su registro y edición, manteniendo actualizada la información de los lugares de acción del voluntariado.

Figura 5
Diagrama de casos de uso Administración del Voluntariado

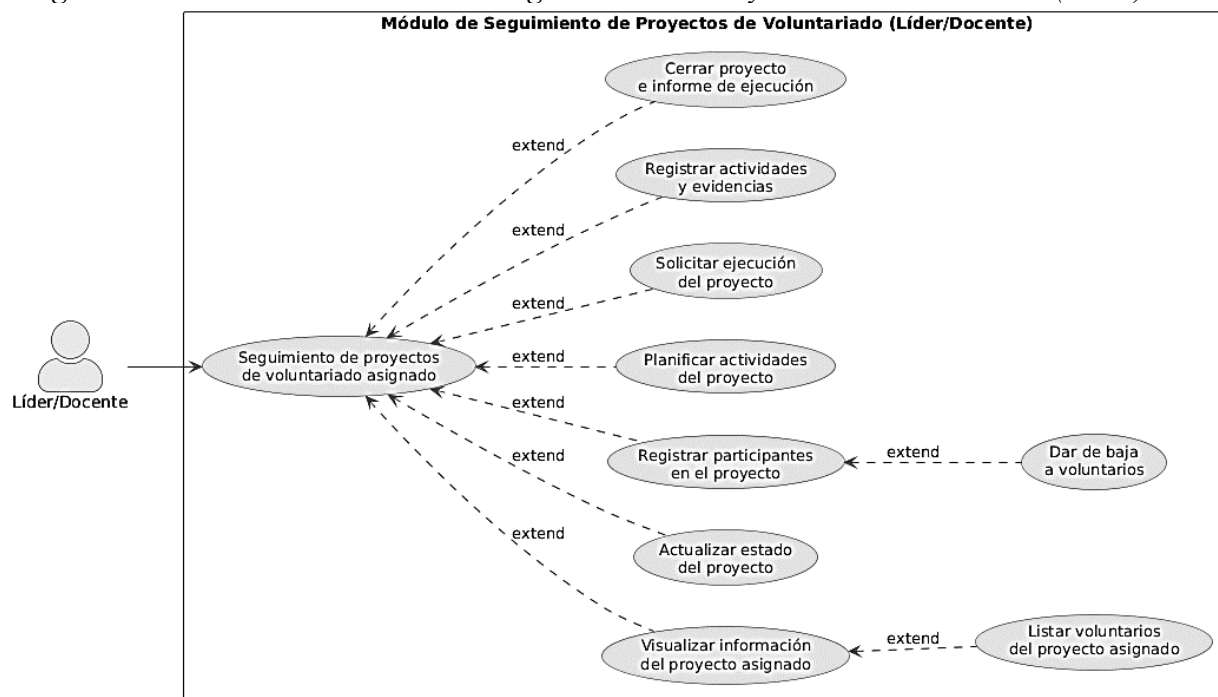


Fuente: Elaboración propia.

Módulo de Gestión de Proyectos de Voluntariado (Líder). El módulo de gestión de proyectos de voluntariado está dirigido a los líderes de los proyectos de voluntariado, responsables de la ejecución operativa de los proyectos de voluntariado asignados. Con este módulo, el líder puede visualizar la información del proyecto asignado, aprobado o rechazado y sus solicitudes de actualizar el estado del proyecto conforme a su avance. Asimismo, permite registrar información relevante del desarrollo del proyecto, garantizando un seguimiento continuo de los participantes y actividades. Como se observa en la Figura 6.

Figura 6

Diagrama de casos de uso Módulo de Seguimiento de Proyectos de Voluntariado (Líder)

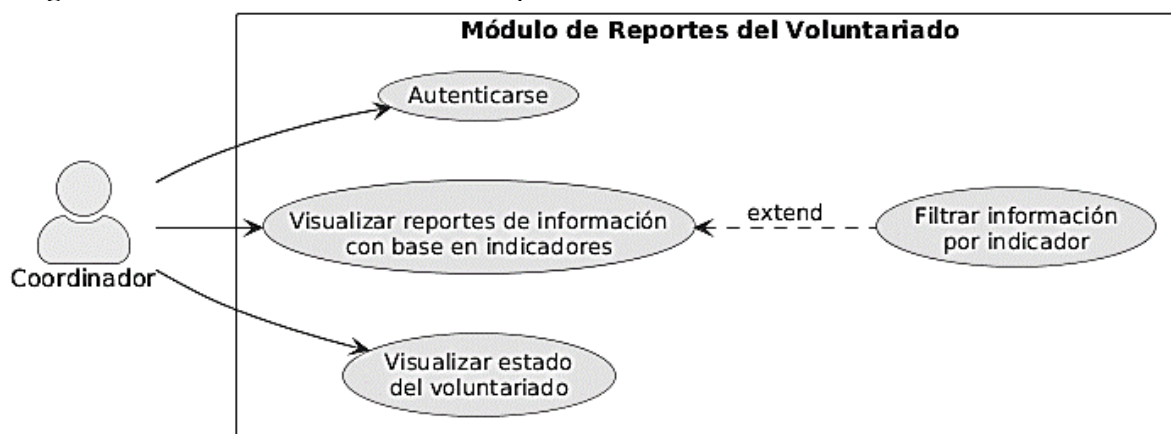


Fuente: Elaboración propia.

Módulo reportes del voluntariado. Este módulo está destinado consultar el listado de proyectos activos, ver participación en su proyecto que posteriormente se utilizarán por otro módulo para la generación de información de trazabilidad. El módulo de seguimiento y reportes permite a los actores autorizados visualizar información consolidada del voluntariado filtrada por reportes específicos. Este módulo facilita el monitoreo del avance de los proyectos y su funcionamiento, sirviendo como apoyo para la supervisión institucional y la toma de decisiones administrativas. Como se observa en la Figura 7.

Figura 7

Diagrama de casos de uso Módulo de reportes del voluntariado



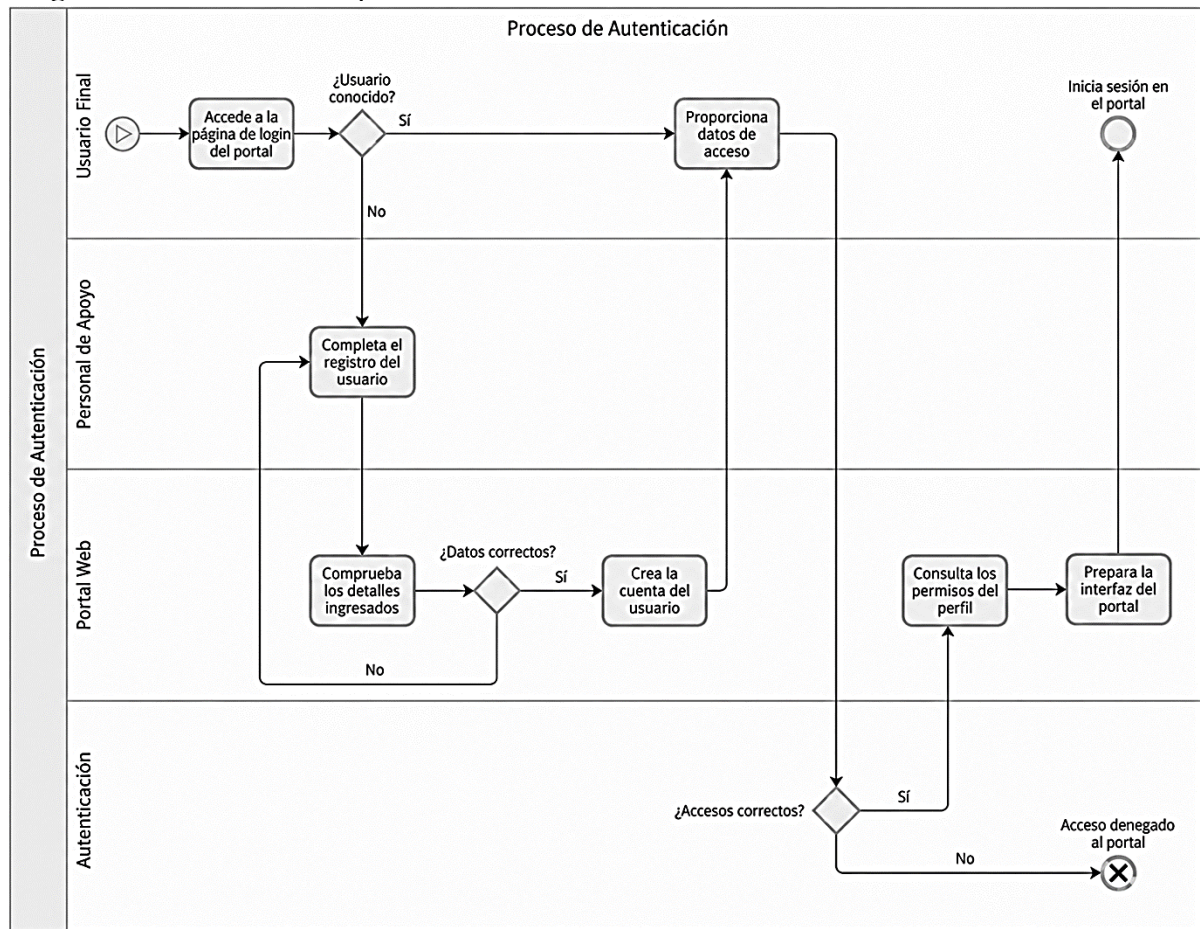
Fuente: Elaboración propia.

En definitiva, se elaboraron los diagramas correspondientes, los cuales permitieron visualizar el comportamiento general de la aplicación y la relación entre los usuarios y las funcionalidades disponibles. Estos diagramas sirvieron como guía para estructurar los módulos del sistema y orientar posteriormente la fase de codificación.

Considerando que el sistema requiere controlar el acceso de los usuarios según el rol asignado, se planteó un proceso de autenticación mediante usuario y credencial autorizada. Este mecanismo permite validar el ingreso al sistema y dirigir a cada usuario hacia las funcionalidades correspondientes, de acuerdo con su perfil dentro de la aplicación.

En este sentido, se elaboró un diagrama de actividades que representa el proceso de inicio de sesión. Dicho diagrama permite observar el flujo que sigue el usuario desde el ingreso de sus credenciales hasta la validación de acceso y la redirección hacia el panel correspondiente. Este proceso se presenta en la Figura 8.

Figura 8
Diagrama de escenario del proceso de autenticación



Fuente: Elaboración propia.

Arquitectura de la aplicación web. La arquitectura de la aplicación web se planteó bajo un modelo en capas, en el cual el frontend y el backend cumplen responsabilidades diferenciadas dentro del sistema. El frontend permite la interacción directa con los usuarios mediante las interfaces web, mientras que el backend se encarga de procesar la lógica de negocio, gestionar las solicitudes del sistema y establecer la comunicación con la base de datos.

Para la organización del sistema se consideró una arquitectura monolítica modular. Esta arquitectura permite mantener las funcionalidades principales dentro de una misma aplicación, pero organizadas en módulos independientes de acuerdo con su responsabilidad. De esta manera, se facilita la administración del código, el mantenimiento del sistema y la incorporación de mejoras futuras.

El backend fue diseñado mediante NestJS, debido a que este framework permite estructurar la aplicación en módulos, controladores y servicios. Esta organización favorece la separación de responsabilidades, ya que cada módulo puede encargarse de una parte específica del sistema,

como autenticación, usuarios, roles, proyectos, áreas de voluntariado, solicitudes, seguimiento y reportes.

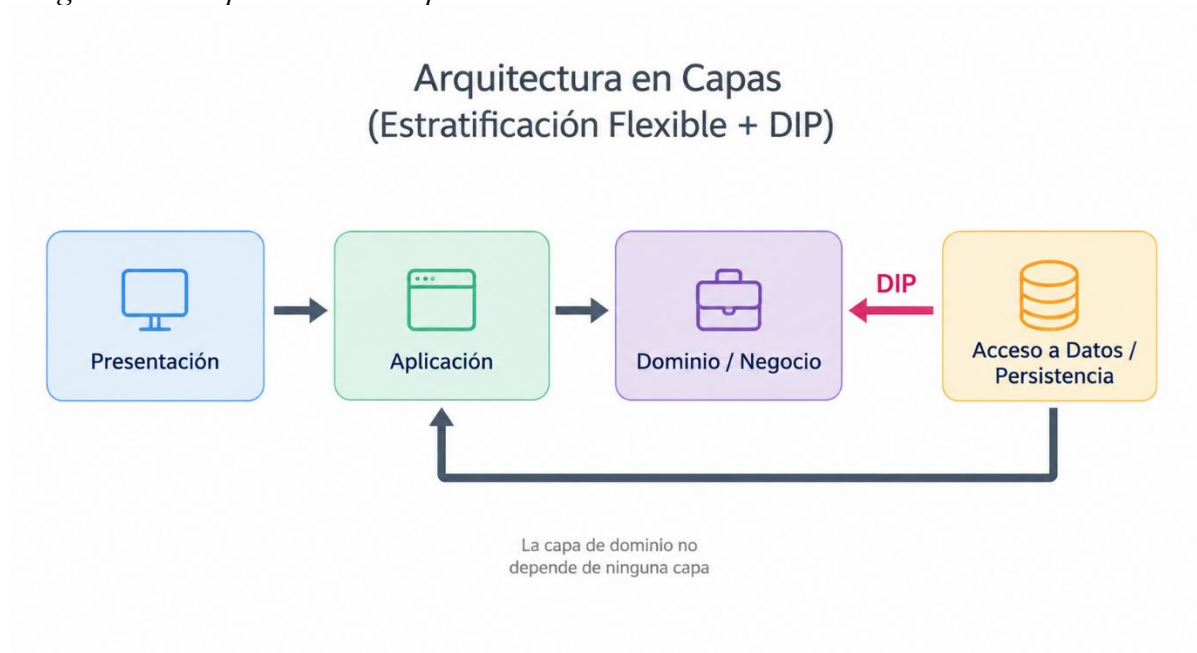
Por su parte, el frontend fue planteado mediante Nuxt, framework basado en Vue.js, con el propósito de construir una interfaz web dinámica y organizada. Esta capa permite que los usuarios interactúen con las funcionalidades del sistema de acuerdo con el rol asignado, facilitando la navegación entre los módulos disponibles.

La comunicación entre el frontend y el backend se realiza mediante servicios internos de la aplicación, permitiendo el envío y recepción de información relacionada con los procesos de voluntariado. Esta estructura permite integrar las operaciones del sistema de manera ordenada, manteniendo una separación entre la presentación de la información y la lógica de procesamiento.

Además, se consideró el uso de Swagger para documentar los servicios desarrollados en el backend. Esta herramienta permite visualizar las rutas disponibles, los parámetros requeridos y las respuestas generadas por el sistema, facilitando la revisión técnica de las funcionalidades implementadas.

La organización arquitectónica propuesta permite que el sistema mantenga una estructura clara, escalable y adecuada para la gestión interna de los proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra, como se muestra en la Figura 9.

Figura 9
Diagrama de Arquitectura en capas del sistema web



Fuente: Elaboración propia.

Diseño de la base de datos. La base de datos, mostrada en la Figura 10, se estructuró mediante tablas relacionales que permiten organizar e interconectar la información utilizada por el sistema web de gestión de proyectos de voluntariado. Este diseño soporta el funcionamiento de la aplicación y contribuye a mantener la integridad, coherencia y disponibilidad de los datos registrados.

Asimismo, el modelo relacional permite centralizar la información del proceso de voluntariado, facilitando su administración y consulta dentro del sistema. Figura 10 presenta las principales relaciones definidas entre las tablas que conforman la base de datos.

Además, la base de datos se programó en SQL, aprovechando las capacidades avanzadas del sistema gestor para optimizar el procesamiento y la administración de la información. Se utilizaron para ello procedimientos almacenados, funciones y consultas especializadas que permiten ejecutar operaciones directamente sobre la base de datos, reduciendo así la carga de procesamiento en las capas superiores de la aplicación y garantizando una mayor eficiencia en la ejecución de las tareas relacionadas con la gestión de proyectos de voluntariado.

También se emplearon transacciones SQL para garantizar la consistencia e integridad de los datos durante operaciones críticas que involucran múltiples registros y tablas relacionadas. Este enfoque permite que las operaciones se realicen de forma rápida, segura y confiable, evitando inconsistencias ante posibles errores durante el procesamiento. Estas avanzadas funcionalidades permiten que la base de datos no sólo sea un repositorio de información, sino también un componente activo de la lógica de gestión y control de procesos del sistema, por consiguiente, se presenta el diseño lógico de la base de datos en la Figura 10.

Diagrama de Base de Datos Voluntariado



Fuente: Elaboración propia.

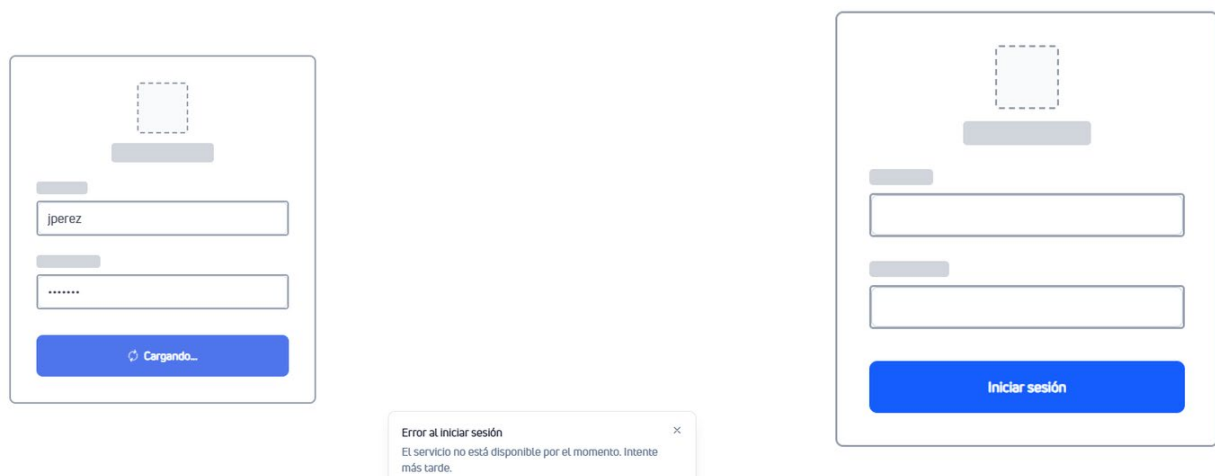
Diseño Interfaz. El diseño de la interfaz del sistema fue concebido bajo principios de usabilidad, claridad y accesibilidad, con el propósito de ofrecer una experiencia de usuario intuitiva y eficiente para los perfiles que interactúan con la aplicación: coordinador, líder de proyecto y usuario voluntario.

La distribución de los elementos visuales se planteó de manera ordenada, procurando que la información sea presentada de forma clara y que las acciones principales puedan identificarse fácilmente. De esta manera, se buscó reducir posibles confusiones durante la navegación y facilitar el uso de las funcionalidades disponibles en el sistema.

Asimismo, las interfaces fueron diseñadas considerando la organización de los módulos y los roles definidos dentro de la aplicación. Esto permite que cada usuario acceda a un entorno visual acorde con sus responsabilidades, manteniendo una estructura coherente entre las diferentes pantallas del sistema.

- **Inicio de sesión.** Como se muestra en la Figura 11, la interfaz de inicio de sesión fue diseñada de forma sencilla, clara y funcional. Esta pantalla permite que el usuario ingrese al sistema mediante credenciales autorizadas, a través de campos destinados al registro del usuario y la contraseña.

Figura 11
Prototipo de visualización responsiva.



Fuente: Elaboración propia.

La disposición de los campos de ingreso y del botón de acceso fue organizada de manera visible y directa, con el fin de facilitar el proceso de autenticación. Además, se consideró una

presentación visual limpia, evitando elementos innecesarios que puedan dificultar la interacción del usuario.

El diseño de esta pantalla responde a criterios básicos de experiencia de usuario, ya que permite que el proceso de ingreso al sistema sea comprensible, rápido y coherente con el resto de la aplicación web.

2.2.3. Fase de codificación

Durante la fase de codificación se llevó a cabo las pautas funcionales del sistema web para la gestión de proyectos de voluntariado, tomando como base las historias de usuario priorizadas en el Product Backlog y los diseños elaborados en la fase anterior. Esta etapa permitió transformar los requerimientos identificados en módulos funcionales dentro de la aplicación, mediante Sprints de desarrollo iterativos e incrementales.

El desarrollo se realizó de manera progresiva, implementando las funcionalidades principales del sistema de acuerdo con los roles establecidos: coordinador, líder de proyecto y usuario voluntario. Esta organización permitió mantener una relación directa entre los requerimientos definidos y las funcionalidades desarrolladas en la plataforma.

Cada módulo fue codificado e integrado siguiendo una estructura organizada, con el propósito de facilitar el mantenimiento del sistema y permitir una mejor separación de responsabilidades. Durante este proceso se implementaron las interfaces de usuario, la lógica de negocio y la conexión con la base de datos, procurando que el sistema responda adecuadamente a los procesos de gestión del voluntariado. Esto permitió identificar ajustes necesarios durante el desarrollo y comprobar que las funcionalidades codificadas mantuvieran correspondencia con las historias de usuario y los diagramas definidos previamente.

Herramientas de desarrollo. Para el desarrollo del sistema web de gestión de proyectos de voluntariado se seleccionaron tecnologías orientadas a la construcción de aplicaciones modernas, organizadas y mantenibles que se desglosan en:

Lenguaje de programación.

- NodeJS: fue utilizado como lenguaje principal para el desarrollo del sistema, debido a que permite trabajar con una estructura más organizada mediante el uso de tipado. Esta característica facilita la detección de errores durante el desarrollo y contribuye a mejorar la mantenibilidad del código.

El uso de NodeJS permitió construir una base más clara para el desarrollo de los diferentes módulos del sistema, favoreciendo la organización de las funcionalidades y la integración entre los componentes de la aplicación web.

Framework para el backend. Para la construcción del lado del servidor se utilizó NestJS, un framework basado en Node.js que permite desarrollar aplicaciones web de forma modular en lenguaje Typescript, con una perspectiva escalable y organizada. Su estructura por módulos, controladores y servicios facilitó la separación de responsabilidades dentro del sistema, permitiendo que cada funcionalidad relacionada con la gestión de proyectos, usuarios, roles, inscripciones y actividades se desarrollara de manera independiente y ordenada.

El uso de NestJS permitió implementar la lógica de negocio del sistema de forma clara, manteniendo una comunicación adecuada entre las solicitudes realizadas desde la interfaz de usuario y los procesos ejecutados en el servidor. Además, al trabajar con una arquitectura orientada a servicios, se facilitó la reutilización de código y el mantenimiento de los diferentes módulos desarrollados durante la fase de codificación.

NestJS permitió aplicar principios de programación orientada a objetos y mantener una organización en capas dentro del servidor. Asimismo, su integración con TypeORM facilitó la conexión con la base de datos y la implementación de controladores, servicios y validaciones.

Base de datos. Para la gestión de la información del sistema se trabajó con una base de datos relacional PostgreSQL, debido a que los datos del sistema mantienen una relación directa entre usuarios, proyectos de voluntariado, roles y actividades. Este modelo permitió organizar la información de forma estructurada y garantizar una mejor integridad de los datos almacenados. La conexión entre el backend y la base de datos permitió realizar las operaciones necesarias para registrar, consultar, actualizar y eliminar información dentro del sistema. De esta manera, las funcionalidades desarrolladas pudieron responder a la información ingresada que desde la aplicación se almacenará correctamente y podrá ser utilizada por los diferentes roles del sistema.

Frontend. La parte visual del sistema fue desarrollada mediante Vue.js, framework utilizado para la construcción de interfaces dinámicas e interactivas. Su uso permitió organizar la aplicación en componentes reutilizables, facilitando el desarrollo de las diferentes vistas del sistema, tales como el inicio de sesión, la gestión de usuarios, la administración de proyectos, la asignación de actividades y la participación de los voluntarios.

Vue.js permitió mejorar la experiencia del usuario al presentar una interfaz más fluida y ordenada, evitando procesos innecesarios de recarga completa de página. Además, el uso de

componentes facilitó la reutilización de elementos visuales dentro del sistema, manteniendo una estructura coherente en las diferentes pantallas de la aplicación web.

Adicionalmente, el desarrollo del frontend se apoyó en Nuxt, framework construido sobre Vue.js que aportó una estructura más ordenada al proyecto mediante el manejo automático de rutas y el renderizado del lado del servidor (SSR), características que contribuyeron a mejorar los tiempos de carga y la fluidez de la navegación. De manera particular, la aplicación se organizó mediante el enfoque de capas (layers) que ofrece Nuxt, el cual permitió estructurar el frontend en niveles independientes y reutilizables, separando la configuración, los componentes y las vistas según su responsabilidad. Esta organización por capas representó un nivel adicional de estructuración en el desarrollo, ya que facilitó el mantenimiento del código, la reutilización de elementos comunes entre los módulos y la incorporación ordenada de nuevas funcionalidades.

Swagger. La comunicación entre NestJS y Nuxt se realizó mediante endpoints estrictamente estructurados con peticiones HTTP, permitiendo que el frontend enviara solicitudes al backend y recibiera respuestas con la información necesaria para mostrarla en la interfaz. Esta integración permitió conectar la lógica del servidor con las acciones realizadas por los usuarios dentro del sistema.

A través de esta comunicación se implementaron las principales funcionalidades del sistema, como registros e inicio de sesión de usuarios, la administración de proyectos, la gestión de seguimiento de la participación de los voluntarios y actividades. Esto permitió que la aplicación funcionara de manera coordinada entre la parte visual, la lógica de negocio y la base de datos. Para documentar estos servicios se integró Swagger, herramienta que generó una documentación técnica interactiva de la API. Mediante esta documentación se especificaron las rutas disponibles, los métodos, los parámetros requeridos y las respuestas esperadas de cada endpoint, lo que facilitó la revisión técnica de las funcionalidades y la ejecución de pruebas de integración directamente desde su interfaz.

Seguridad y autenticación. Con el propósito de garantizar un acceso controlado a las funcionalidades del sistema, se implementaron mecanismos de protección orientados a resguardar las rutas y la información sensible de la aplicación. Por una parte, se emplearon API Keys para restringir el acceso a rutas críticas del backend, de manera que únicamente las solicitudes provenientes de orígenes autorizados pudieran ejecutar operaciones internas o de carácter administrativo.

Por otra parte, la autenticación y autorización de los usuarios se gestionó mediante JSON Web Tokens (JWT), tokens firmados digitalmente que permitieron verificar la identidad del usuario

en cada solicitud realizada al servidor. Este mecanismo posibilitó mantener sesiones seguras dentro de la plataforma y asegurar que cada rol —coordinador, líder o voluntario— accediera únicamente a las funcionalidades correspondientes a sus permisos.

Control de versiones. Se utilizó Gitea, una técnica de control de versiones basada en Git, que permitió mantener un historial detallado de los cambios realizados durante el desarrollo. Su uso facilitó el respaldo del proyecto, el seguimiento de las modificaciones efectuadas en cada módulo y la posibilidad de revertir el código a versiones anteriores ante la presencia de errores. El trabajo se organizó mediante dos ramas principales. La rama principal contuvo la versión estable y validada del sistema, con las funcionalidades integradas y verificadas, mientras que en la rama de desarrollo se realizaron las tareas de codificación, los ajustes y la incorporación de nuevas funcionalidades. Una vez que cada funcionalidad fue completada y probada, los cambios se fusionaron hacia la rama principal, asegurando la estabilidad del sistema durante todo el proceso de desarrollo.

Entorno de desarrollo. Para la codificación del sistema se utilizó Visual Studio Code como editor de código, debido a que ofrece un entorno flexible y compatible con las tecnologías utilizadas en el proyecto. Este editor facilitó la escritura, organización y revisión del código fuente, además de permitir el uso de extensiones que contribuyeron a mejorar el proceso de desarrollo.

2.2.4. Plan de pruebas

Durante la fase de plan de pruebas se diseñó el esquema para verificar el funcionamiento del sistema web para la gestión de proyectos de voluntariado, con la finalidad de comprobar que las funcionalidades desarrolladas respondieran a los requerimientos definidos en las historias de usuario. Este diseño permitió tener la base y plantilla para revisar el comportamiento de los módulos implementados al cierre de cada Sprint y confirmar que las acciones disponibles en la plataforma se ejecutaran de manera correcta de acuerdo con cada rol del sistema.

A continuación, se presentan las plantillas de pruebas funcionales elaboradas para validar las principales historias de usuario del sistema web de gestión de proyectos de voluntariado.

Tabla 7*Plantilla del Plan de Pruebas del Sistema Web de voluntariado*

No.	Historia de usuario
0	-
Campo	-
Escenario	-
Pasos	-
Resultado esperado	-
Resultado obtenido	-
Estado	-
Observaciones	

*Fuente: Elaboración propia.***Tabla 8***Plantilla Prueba funcional — Inicio de sesión del usuario*

No.	Historia de usuario
1	Inicio de sesión del usuario
Campo	Detalle
Escenario	Inicio de sesión con credenciales válidas
Pasos	1. Ingresar correo y contraseña válidos. 2. Presionar el botón Iniciar sesión.
Resultado esperado	El sistema valida las credenciales y redirige al usuario al panel que corresponde a su rol.
Escenario	Inicio de sesión con credenciales incorrectas
Pasos	1. Ingresar correo o contraseña incorrectos. 2. Presionar el botón Iniciar sesión.
Resultado esperado	El sistema niega el acceso y muestra un mensaje indicando que las credenciales no son válidas.
Escenario	Intento de ingreso de un usuario no habilitado
Pasos	1. El usuario de proyecto registra al participante. 2. El recibe acceso al sistema. 3. El ingresa sus credenciales y presiona Iniciar sesión.
Resultado esperado	Él puede acceder al sistema únicamente si fue registrado y habilitado previamente por el coordinador.
Escenario	Intento de ingreso de un rol no habilitado
Pasos	1. Ingresar credenciales de un usuario con un rol que no ha sido habilitado. 2. Presionar Iniciar sesión.
Resultado esperado	El sistema no permite el acceso y muestra un mensaje indicando que el usuario no está habilitado para ingresar.

Fuente: Elaboración propia.

Tabla 9*Plantilla Prueba funcional — Gestión de proyectos de voluntariado*

No.	Historia de usuario
2	Gestión de proyectos de voluntariado
Campo	Detalle
Escenario	Registrar un proyecto con datos válidos
Pasos	1. Ingresar al módulo de proyectos. 2. Completar los datos requeridos del proyecto. 3. Guardar la información.
Resultado esperado	El sistema registra el proyecto correctamente y lo muestra en el listado de proyectos.
Escenario	Modificar la información de un proyecto
Pasos	1. Seleccionar un proyecto existente. 2. Editar los campos necesarios. 3. Guardar los cambios.
Resultado esperado	El sistema actualiza la información del proyecto y muestra los datos modificados.
Escenario	Cambiar el estado de un proyecto
Pasos	1. Seleccionar un proyecto registrado. 2. Cambiar su estado. 3. Confirmar la acción.
Resultado esperado	El sistema actualiza el estado del proyecto y refleja el cambio en la interfaz.

*Fuente: Elaboración propia.***Tabla 10***Plantilla Prueba funcional — Solicitud de ejecución del proyecto*

No.	Historia de usuario
5	Solicitud de ejecución del proyecto
Campo	Detalle
Escenario	Enviar la planificación a revisión
Pasos	1. El líder completa la planificación y las actividades del proyecto en estado Asignado responsable. 2. Selecciona la opción Enviar a revisión. 3. Confirma la acción.
Resultado esperado	El sistema cambia el estado del proyecto a Enviado a revisión y lo muestra como pendiente en el panel del coordinador.
Escenario	Reenviar la planificación tras un rechazo
Pasos	1. El líder ingresa a un proyecto en estado Rechazado. 2. Corrige la planificación según el motivo indicado por el coordinador. 3. Selecciona la opción Reenviar.
Resultado esperado	El sistema cambia el estado del proyecto a Reenviado y lo muestra nuevamente como pendiente en el panel del coordinador.
Escenario	Intentar enviar a revisión un proyecto en un estado no habilitado
Pasos	1. El líder ingresa a un proyecto que no se encuentra en estado Asignado responsable ni Rechazado (por ejemplo, ya Aprobado). 2. Intenta acceder a la opción de enviar o reenviar.
Resultado esperado	El sistema no habilita la opción de enviar/reenviar mientras el proyecto no se encuentre en un estado que lo permita.

Fuente: Elaboración propia.

Tabla 11*Plantilla Prueba funcional — Registro de participantes voluntarios*

No.	Historia de usuario
3	Registro de participantes voluntarios internos
Campo	Detalle
Escenario	Registrar un participante con datos válidos
Pasos	1. El líder de proyecto ingresa al módulo de participantes. 2. Completa los datos del voluntario. 3. Guarda la información.
Resultado esperado	El sistema registra al participante correctamente y lo muestra en el listado correspondiente.
Escenario	Intentar registrar participantes voluntarios externos
Pasos	1. Ingresar los datos de un voluntario ya registrado. 2. Guardar la información.
Resultado esperado	El sistema detecta la duplicidad y muestra un mensaje indicando que el participante ya existe.
Escenario	Revisar información de participantes
Pasos	1. Seleccionar un participante registrado. 2. Activar la opción de acceso a los datos. 3. Guardar los cambios.
Resultado esperado	El sistema habilita el acceso a ver la información de participante, permitiéndole editar en caso de ser externo.

*Fuente: Elaboración propia.***Tabla 12***Plantilla Prueba funcional — Planificación de actividades del proyecto*

No.	Historia de usuario
4	Planificación de actividades del proyecto
Campo	Detalle
Escenario	Registrar una nueva actividad
Pasos	1. Ingresar al módulo de actividades. 2. Completar los datos de la actividad. 3. Guardar la información.
Resultado esperado	El sistema registra la actividad y la asocia correctamente al proyecto correspondiente.
Escenario	Modificar una actividad registrada
Pasos	1. Seleccionar una actividad existente. 2. Editar la información necesaria. 3. Guardar los cambios.
Resultado esperado	El sistema actualiza la información de la actividad y muestra los cambios realizados.
Escenario	Eliminar o desactivar una actividad
Pasos	1. Seleccionar una actividad registrada. 2. Elegir la opción de eliminar o desactivar. 3. Confirmar la acción.
Resultado esperado	El sistema elimina o desactiva la actividad seleccionada y actualiza el listado mostrado.

Fuente: Elaboración propia.

Tabla 13*Plantilla Prueba funcional — Autorización del coordinador*

No.	Historia de usuario
6	Autorización del cambio de estado por el coordinador
Campo	Detalle
Escenario	Aprobar un proyecto enviado a revisión o reenviado
Pasos	1. El coordinador ingresa a la planificación de un proyecto en estado Enviado a revisión o Reenviado. 2. Selecciona la opción Aprobar. 3. Confirma la decisión.
Resultado esperado	El sistema actualiza el estado del proyecto a Aprobado y lo habilita para que el líder registre evidencias de ejecución.
Escenario	Rechazar un proyecto indicando un motivo
Pasos	1. El coordinador ingresa a la planificación del proyecto. 2. Selecciona la opción Rechazar. 3. Ingresa un motivo de rechazo y confirma.
Resultado esperado	El sistema actualiza el estado del proyecto a Rechazado y almacena el motivo ingresado, visible para el líder.
Escenario	Reenviar un proyecto rechazado tras corregir la planificación
Pasos	1. El líder ingresa a un proyecto en estado Rechazado. 2. Corrige la planificación observada. 3. Envía nuevamente la solicitud.
Resultado esperado	El sistema actualiza el estado del proyecto a Reenviado y lo muestra como pendiente de revisión en el panel del coordinador.

*Fuente: Elaboración propia.***Tabla 14***Plantilla Prueba funcional — Registro de actividades realizadas y evidencias*

No.	Historia de usuario
7	Registro de actividades realizadas y evidencias
Campo	Detalle
Escenario	Subir una evidencia en formato PDF
Pasos	1. El líder ingresa a un proyecto en estado Aprobado o Iniciado. 2. Selecciona la opción de subir evidencia. 3. Adjunta un archivo PDF y guarda.
Resultado esperado	El sistema registra la evidencia y la muestra en el listado del proyecto correspondiente.
Escenario	Intentar subir una evidencia en un formato no permitido
Pasos	1. El líder selecciona la opción de subir evidencia. 2. Intenta adjuntar un archivo que no sea PDF.
Resultado esperado	El sistema rechaza el archivo y muestra un mensaje indicando que solo se permiten archivos en formato PDF.
Escenario	Intentar subir evidencia en un estado no permitido
Pasos	1. El líder ingresa a un proyecto que no se encuentra en estado Aprobado ni Iniciado. 2. Intenta acceder a la opción de subir evidencia.
Resultado esperado	El sistema no habilita la opción de subir evidencia mientras el proyecto no se encuentre en un estado permitido.

Fuente: Elaboración propia.

Tabla 15*Plantilla Prueba funcional — Cierre del proyecto e informe de ejecución*

No.	Historia de usuario
8	Cierre del proyecto e informe de ejecución
Campo	Detalle
Escenario	Finalizar un proyecto adjuntando el informe final
Pasos	1. El líder ingresa a un proyecto en estado Aprobado. 2. Selecciona la opción Finalizar proyecto. 3. Adjunta el informe final en PDF y confirma.
Resultado esperado	El sistema actualiza el estado del proyecto a Finalizado y almacena el informe adjuntado.
Escenario	Intentar finalizar un proyecto sin adjuntar el informe
Pasos	1. El líder selecciona la opción Finalizar proyecto. 2. Intenta confirmar la acción sin adjuntar el informe en PDF.
Resultado esperado	El sistema no permite confirmar la acción y muestra un mensaje indicando que el informe es obligatorio.
Escenario	Consultar el informe final de un proyecto finalizado
Pasos	1. El usuario ingresa al listado de proyectos. 2. Selecciona la opción Ver informe en un proyecto en estado Finalizado.
Resultado esperado	El sistema muestra el informe final almacenado correspondiente al proyecto.

Fuente: Elaboración propia.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos a partir de la implementación del sistema web para la gestión de proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra. En primer lugar, se describen las interfaces construidas para cada uno de los roles del módulo, destacando los criterios de usabilidad aplicados; posteriormente, se expone la validación funcional del sistema mediante la ejecución del plan de pruebas definido en el capítulo anterior; finalmente, se discuten los resultados alcanzados en función de los objetivos planteados.

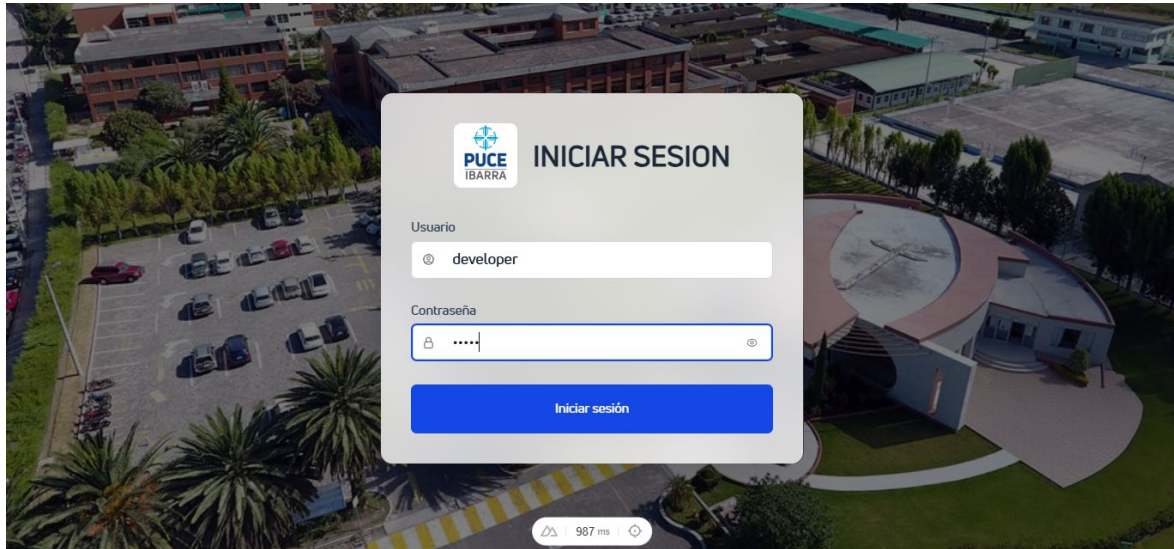
3.1. Interfaces del sistema

Las interfaces del sistema de voluntariado se desarrollaron con Nuxt, reutilizando componentes bajo el concepto de Nuxt UI y aplicando estilos que garantizan una visualización responsiva y coherente en toda la plataforma. El diseño priorizó la experiencia de usuario mediante una navegación clara, tablas con búsqueda y filtros, y ventanas modales para las operaciones de registro y edición, respetando los flujos de trabajo propios de cada rol. La comunicación con el backend se realiza de forma continua mediante endpoints documentados con Swagger/OpenAPI, lo que garantiza una integración estructurada y verificable entre ambas capas.

El acceso al módulo se realiza mediante la autenticación institucional del sistema de voluntariado, que valida las credenciales a través de la API de la universidad utilizando el Directorio Activo (LDAP), como se muestra en la Figura 12; una vez validadas, la plataforma habilita únicamente las opciones de menú correspondientes al rol del usuario, de manera que el coordinador y el líder de proyecto ingresan a entornos de trabajo diferenciados.

Figura 12

Inicio de sesión del sistema de voluntariado



Fuente: Elaboración propia.

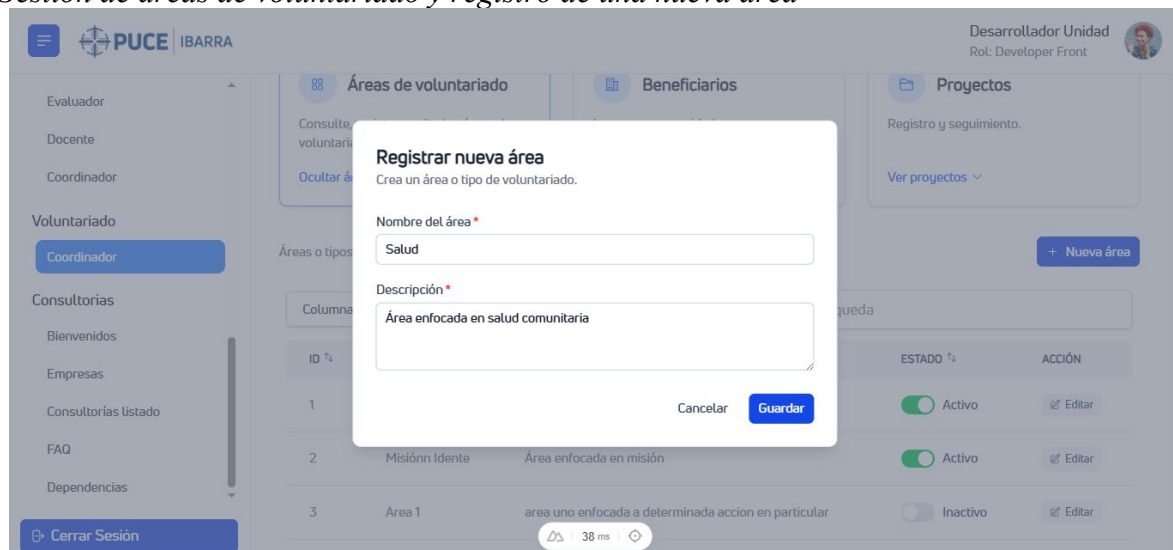
La Figura 12 muestra la pantalla de inicio de sesión del sistema de voluntariado, mediante la cual los usuarios ingresan con sus credenciales institucionales. La autenticación se valida contra el Directorio Activo (LDAP) a través de la API de la universidad, y el sistema determina las rutas y opciones de menú disponibles según el rol asignado.

3.1.1. Módulo de administración del voluntariado (Coordinador)

El módulo del coordinador concentra la administración general del voluntariado en un panel organizado en tres ámbitos: áreas de voluntariado, beneficiarios y proyectos. A continuación, se presentan sus interfaces siguiendo el flujo de gestión de un proyecto.

Figura 13

Gestión de áreas de voluntariado y registro de una nueva área

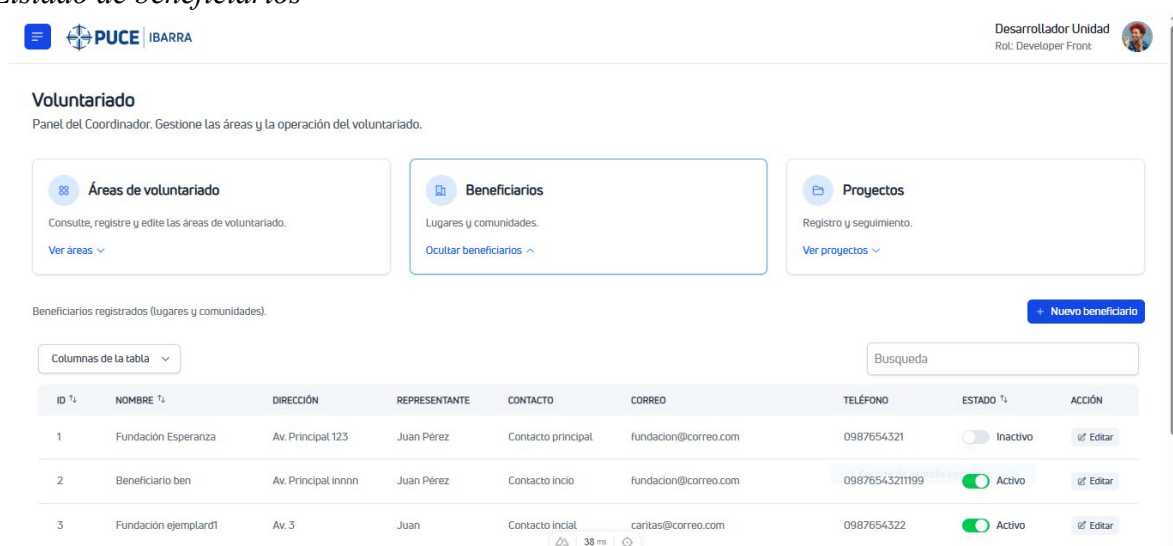


Fuente: Elaboración propia.

La Figura 13 muestra la sección de áreas de voluntariado con la ventana de registro de una nueva área, que solicita el nombre y la descripción de la categoría. El estado de cada área se administra mediante un interruptor de activación, lo que permite conservar el historial sin eliminar registros físicamente.

Figura 14

Listado de beneficiarios



Fuente: Elaboración propia.

La Figura 14 presenta el listado de beneficiarios, entidades o lugares de acción donde se ejecutan los proyectos. La vista incorpora búsqueda, selección de columnas y acceso a la edición de cada registro.

Figura 15
Registro de un beneficiario

Registrar nuevo beneficiario
Registra un beneficiario (lugar o comunidad).

Nombre *
Ej. Fundación Esperanza
El nombre es requerido

Teléfono *
Ej. 0987654321
El teléfono es requerido

Representante
Ej. Juan Pérez

Contacto
Ej. Contacto principal

Correo electrónico
Ej. fundacion@correo.com

Dirección
Ej. Av. Principal 123

Desarrollador Unidad
Rol: Developer Front

Voluntariado
Panel del Coordinador. Gestione las áreas y la operación del voluntariado.

Áreas de voluntariado
Consulte, registre y edite las áreas de voluntariado.
Ver áreas

Beneficiarios registrados (lugares y comunidades).

Columnas de la tabla

ID	NOMBRE	DIRECCIÓN
1	Fundación Esperanza	Av. Principal 123
2	Beneficiario ben	Av. Principal innnn
3	Fundación ejemplard1	Av. 3

Proyectos
Registro y seguimiento.
Ver proyectos

Busqueda

TELÉFONO	ESTADO	ACCIÓN
0987654321	Inactivo	Editar
09876543211199	Activo	Editar
0987654322	Activo	Editar

+ Nuevo beneficiario

Fuente: Elaboración propia.

La Figura 15 muestra la ventana de registro de beneficiarios, con el nombre y el teléfono como datos obligatorios, complementados con la dirección y una descripción del lugar de acción.

Figura 16
Gestión de proyectos de voluntariado con filtros por estado

Desarrollador Unidad
Rol: Developer Front

Gestión de proyectos de voluntariado

Historial de ciudades
Gestión de citas
Reporte de citas
Mantenimiento
Reporte de inventario
Mantenimiento preventivo
Administración de catálogos
Inventario
Proyectos de investigación
Evaluador
Docente
Coordinador
Voluntariado
Consultorías
Bienvenidos
Empresas
Consultorías listado
FAQ
Dependencias

Cerrar Sesión

Todos Creado Asignado responsable Enviado a revisión Rechazado Aprobado Finalizado

Columnas de la tabla

Busqueda

ID	PROYECTO	DESCRIPCIÓN	BENEFICIARIO	LÍDER	ESTADO	CREACIÓN	ACCIÓN
1	Proyecto Solidario prouebam	Descripción detalladam	Fundación Esperanza	Juan Libarra	Asignado responsable	3 jul 2026, 7:32 a. m.	Ver participantes Ver planificación Editar líder
2	Proyecto salud_esperanza	Descripción Proyecto salud_esperanza	Fundación Esperanza	DAVID RUMINAHUI GUANDINANGO CONEJO	Finalizado	3 jul 2026, 10:58 a. m.	Ver participantes Ver planificación Ver evidencias Ver informe
3	proyecto ocean	Descripción ocean	Beneficiario ben	DAVID RUMINAHUI GUANDINANGO CONEJO	Finalizado	3 jul 2026, 10:41 a. m.	Ver participantes Ver planificación Ver evidencias Ver informe
4	Proyectoooo	Descripción Proyectoooo	Beneficiario ben	CAROLINA MISHIEL ANRANGO ALBA	Enviado a revisión	3 jul 2026, 7:47 a. m.	Ver participantes Ver planificación Editar líder Aprobar Rechazar

38 ms

Fuente: Elaboración propia.

La Figura 16 corresponde a la gestión de proyectos, interfaz central del coordinador. Los chips superiores filtran por estado según la máquina de estados del proyecto, y las acciones de cada fila se habilitan dinámicamente: revisión de participantes y planificación, edición del líder, consulta de evidencias e informe en los finalizados, y decisión de aprobación en los enviados a revisión.

Figura 17
Registro de un nuevo proyecto

The image shows a web application interface for project management. A modal window titled "Registrar nuevo proyecto" is open, allowing the user to create a new project. The modal contains the following fields and elements:

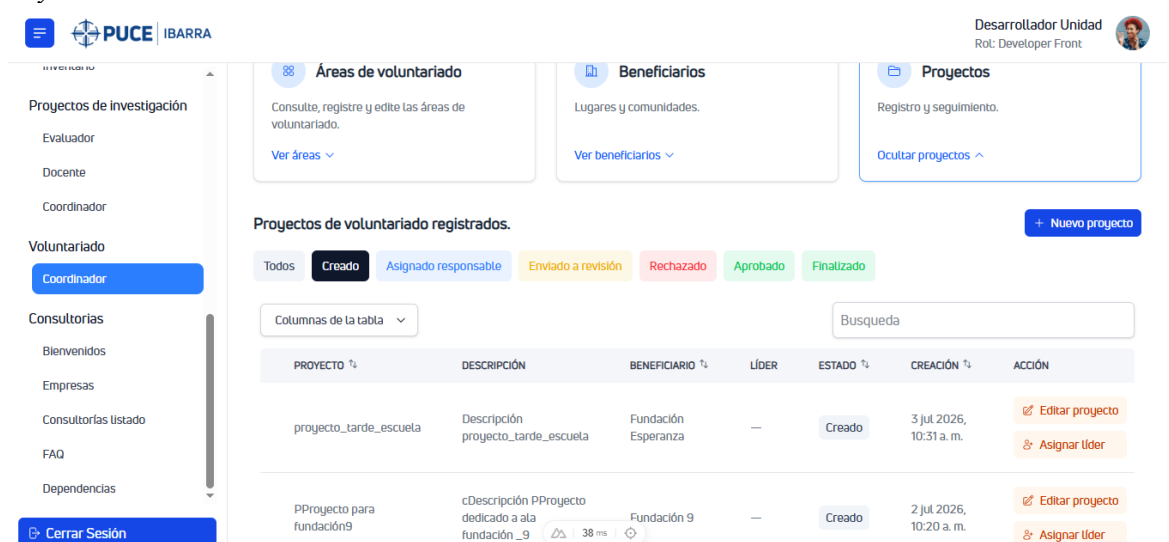
- Área ***: A dropdown menu with "Salud" selected.
- Beneficiario ***: A dropdown menu with "Fundación A" selected.
- Nombre ***: A text input field containing "Ej. Proyecto Solidario". A red error message "El nombre es requerido" is displayed below it.
- Descripción ***: A text input field containing "Descripción del proyecto". A red error message "La descripción es requerida" is displayed below it.
- Observación**: A text input field.
- Buttons**: "Cancelar" and "Guardar" buttons at the bottom right of the modal.

The background interface shows a sidebar menu with options like "Historial de ciudadanos", "Gestión de citas", "Reporte de citas", "Mantenimiento", "Reporte de inventario", "Mantenimiento preventivo", "Administración de catálogos", "Inventario", "Proyectos de investigación", "Evaluador", "Docente", "Coordinador", and "Voluntariado". The main content area is titled "Voluntariado" and includes a "Panel del Coordinador" and a "Proyectos de voluntariado" table. A "Proyectos" section on the right has a "Registro y seguimiento" button and a "Ocultar proyectos" button.

Fuente: Elaboración propia.

La Figura 17 muestra la ventana de registro de un proyecto, que vincula el área de voluntariado y el beneficiario, junto con el nombre y la descripción. El proyecto se crea en el estado inicial Creado.

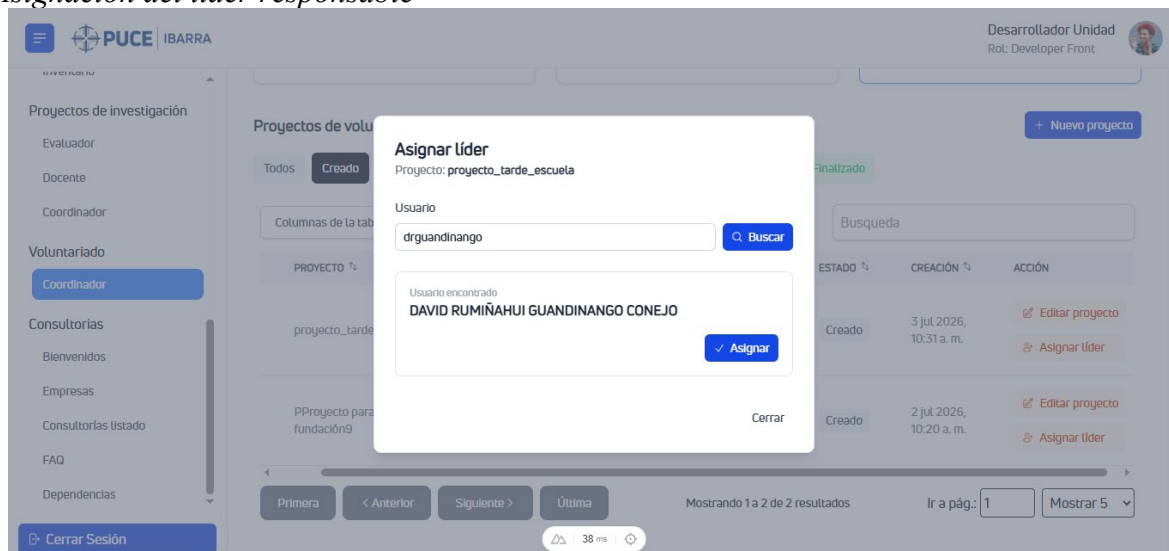
Figura 18
Proyecto en estado Creado



Fuente: Elaboración propia.

La Figura 18 presenta un proyecto recién registrado. En el estado creado el sistema habilita únicamente la edición del proyecto y la asignación del líder responsable, pues aún no existe información operativa que revisar.

Figura 19
Asignación del líder responsable



Fuente: Elaboración propia.

La Figura 19 muestra la ventana de asignación del líder: el coordinador busca al usuario institucional y el sistema autocompleta su nombre. Con la asignación, el proyecto pasa al estado Asignado responsable y aparece en el panel del líder.

Figura 20

Proyecto enviado a revisión con las acciones de decisión

The screenshot displays the PUCE Ibarra web application interface. On the left is a sidebar menu with categories like 'Bienestar Universitario', 'Laboratorios reservas', 'Laboral', 'Citas - KUNA Contigo', and 'Mantenimiento'. The top navigation bar includes tabs for 'Áreas de voluntariado', 'Beneficiarios', and 'Proyectos'. The main content area shows 'Proyectos de voluntariado registrados.' with a table of projects. The project 'proyecto_tarde_escuela' is in the 'Enviado a revisión' status, indicated by a yellow tag. To the right of the project row are buttons for 'Ver participantes', 'Ver planificación', 'Editar líder', 'Aprobar' (green), and 'Rechazar' (red). The bottom of the page shows a footer with 'Cerrar Sesión' and a status bar.

Fuente: Elaboración propia.

La Figura 20 muestra un proyecto en estado Enviado a revisión, con los botones de Aprobar y Rechazar destacados en colores sólidos que señalan la decisión pendiente del coordinador.

Figura 21

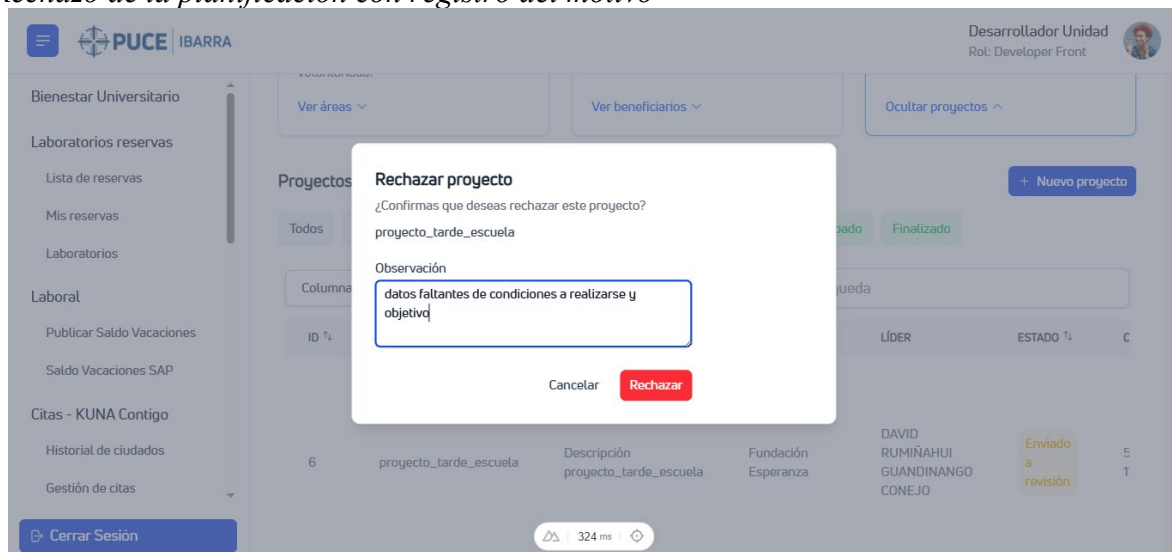
Revisión de la planificación enviada

The screenshot shows the 'Planificación' modal window in the PUCE Ibarra web application. The window title is 'Planificación' and the subtitle is 'proyecto_tarde_escuela'. It contains a question '¿Deseas aprobar esta planificación?' with 'Rechazar' (red) and 'Aprobar' (green) buttons. Below this is a 'Resumen' section with 'Actividades' listed in a table. The table has columns: 'NECESIDAD', 'ACCIÓN PROPUESTA', 'RECURSOS', 'FECHA EJECUCIÓN', and 'ACCIONES'. The activities listed are 'Taller creativo', 'Curiosidad científica', and 'Expresión oral y trabajo en equipo'. The 'Aprobar' button is highlighted in green.

Fuente: Elaboración propia.

La Figura 21 presenta la ventana de revisión de la planificación, donde el coordinador consulta el resumen, el objetivo y las actividades registradas por el líder antes de decidir; la decisión se ejecuta desde esta misma vista, garantizando una revisión informada.

Figura 22
Rechazo de la planificación con registro del motivo



Fuente: Elaboración propia.

La Figura 22 muestra la confirmación del rechazo, que permite registrar una observación con el motivo de la decisión; esta observación queda visible para el líder, quien puede corregir la planificación y reenviarla.

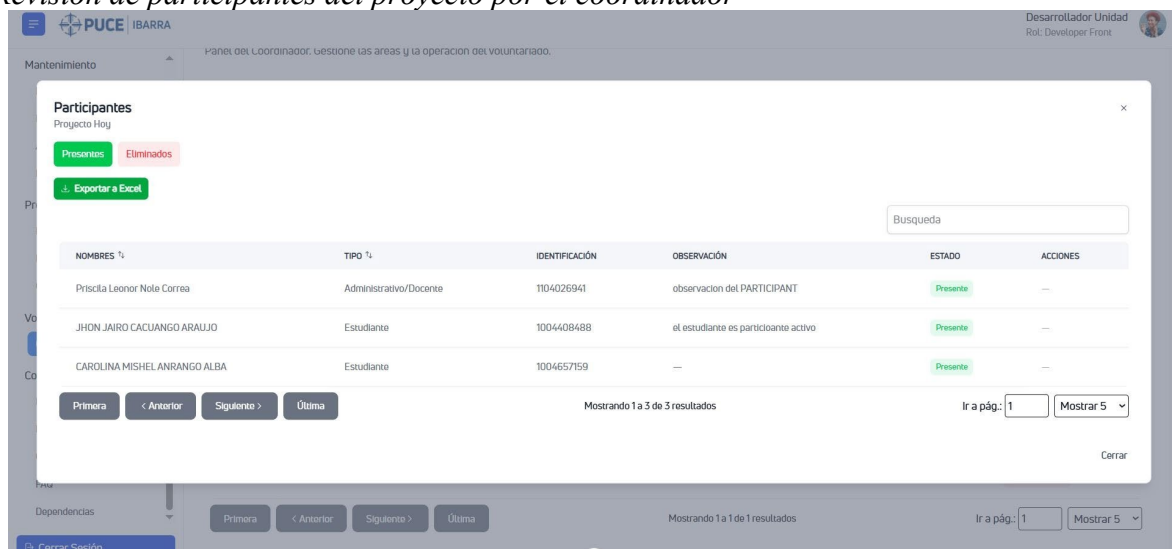
Figura 23
Proyecto reenviado con la decisión pendiente



Fuente: Elaboración propia.

La Figura 23 presenta el proyecto en estado Reenviado después de que el líder aplicó las correcciones. El coordinador vuelve a disponer de las acciones de decisión; con la aprobación, el proyecto pasa al estado Aprobado y se habilita la ejecución.

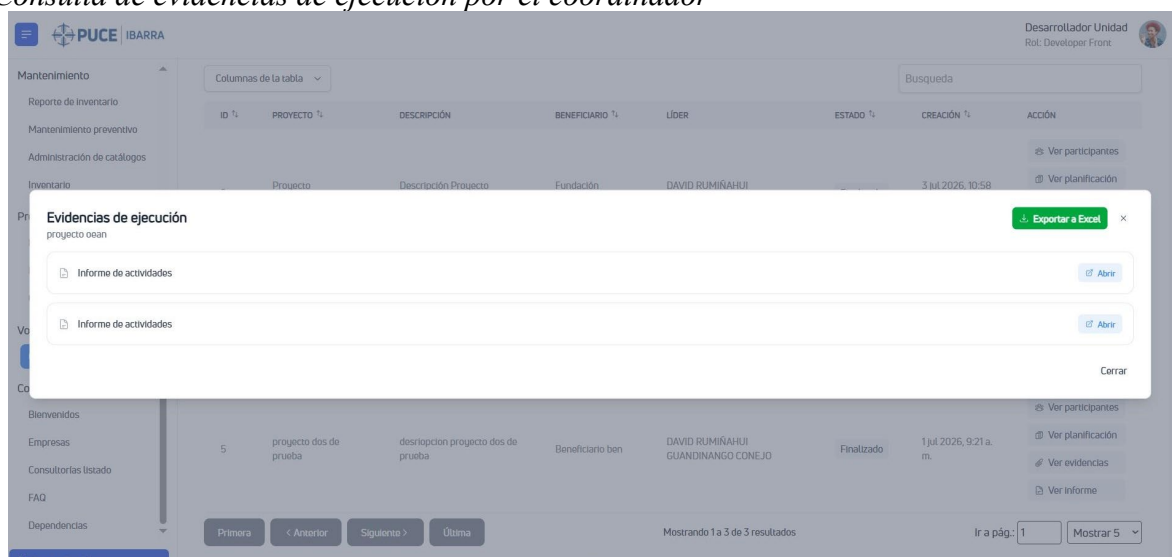
Figura 24
Revisión de participantes del proyecto por el coordinador



Fuente: Elaboración propia.

La Figura 24 presenta la ventana de revisión de participantes, de solo lectura para el coordinador, con el tipo de voluntario, la identificación, las observaciones y el estado de cada participante, además de la exportación a Excel.

Figura 25
Consulta de evidencias de ejecución por el coordinador



Fuente: Elaboración propia.

La Figura 25 muestra la ventana de evidencias de ejecución desde la vista del coordinador, con los documentos cargados por el líder disponibles para su apertura; en los proyectos finalizados se habilita además la consulta del informe final.

3.1.2. Módulo de gestión de proyectos de voluntariado (Líder)

El módulo del líder de proyecto está orientado a la operación de los proyectos asignados, desde el registro de participantes y la elaboración de la planificación hasta la carga de evidencias y el cierre formal con el informe final.

Figura 26

Panel del líder con un proyecto asignado

Voluntariado
Panel del Líder. Gestione la operación asignada a su liderazgo.

Proyectos asignados a su liderazgo.

Todos **Asignado responsable** Rechazado Aprobado Finalizado

Columnas de la tabla

PROYECTO ¹	DESCRIPCIÓN	BENEFICIARIO ¹	ESTADO	PARTICIPANTES	ACTIVIDADES
proyecto_tarde_escuela	Descripción proyecto_tarde_escuela	Fundación Esperanza	Asignado responsable	+ Registrar	+ Registrar

Primera < Anterior Siguiente > Última

Mostrando 1 a 1 de 1 resultados Ir a pág.: Mostrar 5

Cerrar Sesión

340 ms

Fuente: Elaboración propia.

La Figura 26 muestra el panel del líder con un proyecto en estado Asignado responsable y los accesos directos a la gestión de participantes y actividades; los chips superiores permiten filtrar los proyectos por estado.

Figura 27
Registro de un participante del proyecto

Fuente: Elaboración propia.

La Figura 27 presenta la ventana de registro de participantes, que distingue tres tipos de voluntario (administrativo o docente, estudiante y externo). Para los perfiles institucionales el sistema busca al usuario y autocompleta su identificación y nombre; los externos se ingresan de forma manual con una observación.

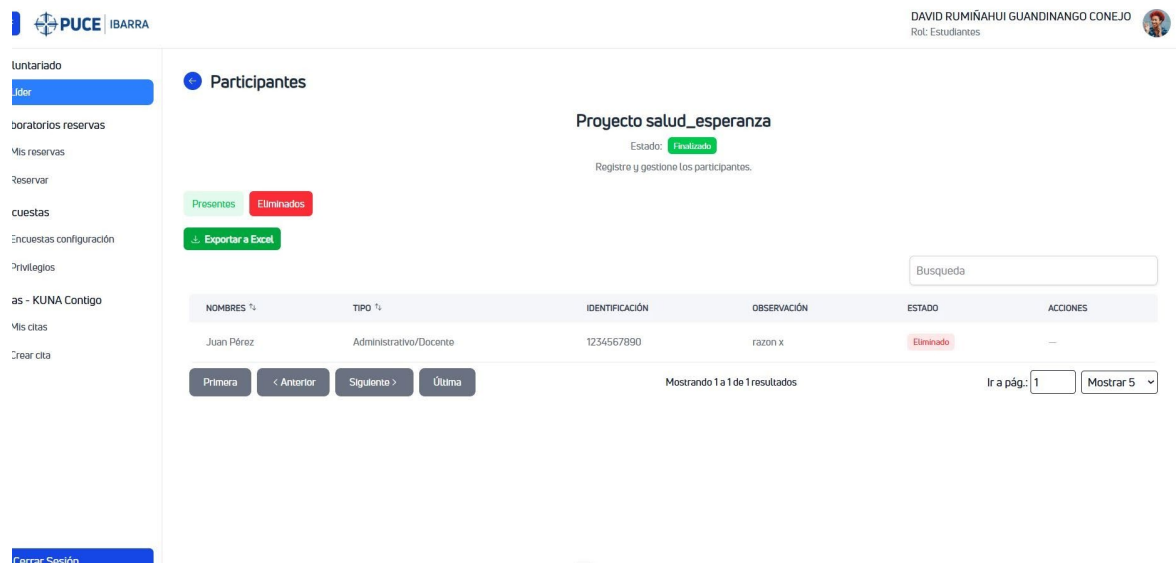
Figura 28
Listado de participantes presentes

NOMBRES	TIPO	IDENTIFICACIÓN	OBSERVACIÓN	ESTADO	ACCIONES
Luis Sánchez	Externo	1002003001	—	Presente	Editar Eliminar
Alfredo Bonilla	Externo	1005042778	—	Presente	Editar Eliminar
Manuel Lita	Externo	1002004004	—	Presente	Editar Eliminar

Fuente: Elaboración propia.

La Figura 28 muestra el listado de participantes presentes del proyecto, con su tipo, identificación y estado, la exportación a Excel y los filtros de presentes y eliminados.

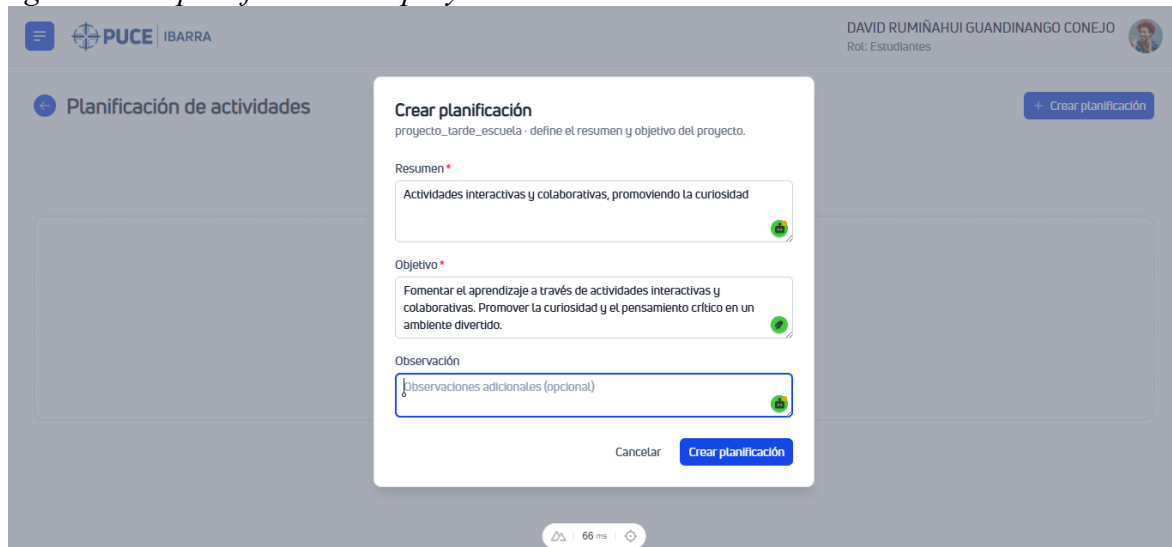
Figura 29
Historial de participantes eliminados



Fuente: Elaboración propia.

La Figura 29 muestra el filtro de participantes eliminados. La eliminación exige registrar un motivo que se conserva como observación, de modo que el proyecto mantiene el historial completo de su nómina incluso después del cierre.

Figura 30
Registro de la planificación del proyecto



Fuente: Elaboración propia.

La Figura 30 presenta la ventana de registro de la planificación, con el resumen y el objetivo del plan de trabajo. La planificación permanece editable únicamente en los estados Asignado responsable o Rechazado.

Figura 31
Actividades registradas de la planificación

PUCE IBARRA

DAVID RUMINAHUI GUANDINANGO CONEJO
Rol: Estudiantes

Planificación de actividades

Editar planificación Enviar a revisión

Resumen: Actividades interactivas y colaborativas, promoviendo el aprendizaje a través de actividades interactivas.
Objetivo: Fomentar el aprendizaje a través de actividades interactivas.

Actividades

Exportar a Excel

NECESIDAD % ACCIÓN PROPUESTA

Registrar actividad
proyecto_tarde_escuela

Necesidad detectada *
Taller creativo.

Acción propuesta *
Taller de Collage

Recursos *
Tijeras, hojas, colores

Fecha de ejecución *
05/07/2026

Cancelar Registrar

Busqueda

ACCIONES

Primera < Anterior Siguiente > Última

66 ms 0 de 0 resultados

Ir a pág.: 1 Mostrar 5

Fuente: Elaboración propia.

La Figura 31 muestra la pantalla de actividades del proyecto, cada una con la necesidad detectada, la acción propuesta, los recursos y la fecha de ejecución, junto con la exportación a Excel.

Figura 32
Envío de la planificación a revisión

PUCE IBARRA

DAVID RUMINAHUI GUANDINANGO CONEJO
Rol: Estudiantes

Planificación de actividades

Editar planificación Enviar a revisión

proyecto_tarde_escuela

Estado: Asignado responsable

Resumen: Actividades interactivas y colaborativas, promoviendo el aprendizaje a través de actividades interactivas.
Objetivo: Fomentar el aprendizaje a través de actividades interactivas.

Actividades

Exportar a Excel

NECESIDAD % ACCIÓN PROPUESTA RECURSOS FECHA EJECUCIÓN % ACCIONES

Taller creativo. Taller de Collage Tijeras, hojas, colores 2026-07-05 Editar Eliminar

Curiosidad científica. Taller de mini-experimentos (volcán de bicarbonato) 2026-07-05 Editar Eliminar

Enviar a revisión

¿Confirmas que deseas enviar este proyecto a revisión del coordinador?

Cancelar Confirmar

Busqueda

ACCIONES

Primera < Anterior Siguiente > Última

66 ms 0 de 0 resultados

Ir a pág.: 1 Mostrar 5

Fuente: Elaboración propia.

La Figura 32 presenta la confirmación del envío a revisión. Con esta acción el proyecto pasa al estado Enviado a revisión y queda pendiente de la decisión del coordinador.

Figura 33

Planificación rechazada con el motivo visible y opción de reenvío

The screenshot shows the 'Planificación de actividades' (Activity Planning) interface. At the top, the user is identified as DAVID RUMINAHUI GUANDINANGO CONEJO, Rol: Estudiantes. The main heading is 'planificación de actividades'. Below it, the project name 'proyecto_tarde_escuela' is displayed with a status of 'Rechazado' (Rejected). A red banner indicates the reason: 'Proyecto rechazado: datos faltantes de condiciones a realizarse y objetivo' (Project rejected: missing data on conditions to be carried out and objective). Below this, a summary and objective are provided. The 'Actividades' section includes a table with columns: NECESIDAD, ACCIÓN PROPUESTA, RECURSOS, FECHA EJECUCIÓN, and ACCIONES. The table lists activities like 'Taller creativo' and 'Taller de Collage'. A 'Subir evidencia' (Upload evidence) button is visible in the bottom right corner.

Fuente: Elaboración propia.

La Figura 33 muestra la planificación de un proyecto rechazado: el sistema presenta un mensaje fijo con el motivo registrado por el coordinador y habilita la opción Reenviar para someter nuevamente la planificación corregida.

Figura 34

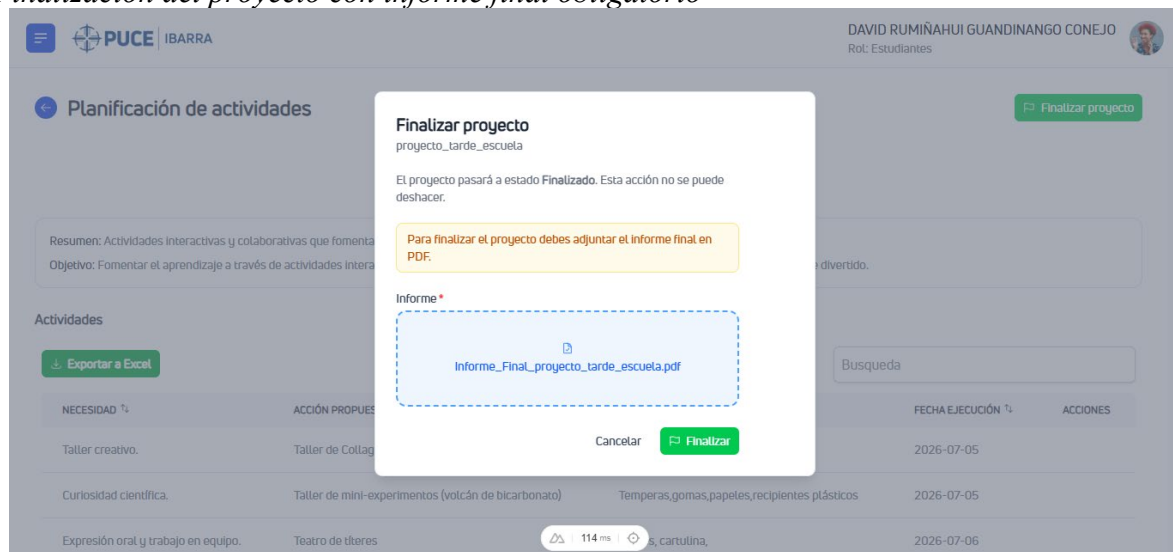
Registro de evidencias de ejecución en formato PDF

The screenshot shows the 'Subir evidencia' (Upload evidence) modal form. The form is titled 'Subir evidencia' and has a subtitle 'proyecto_tarde_escuela'. It contains a 'Nombre / descripción' field with the value 'Evidencia_actividad_taller_creativo'. Below this is an 'Archivo' field with a dashed box indicating the upload area. The file name 'Evidencia_actividad_taller_creativo.pdf' is visible. At the bottom of the modal, there are 'Cancelar' and 'Subir' buttons. The background shows the same interface as Figure 33, but with the modal form open.

Fuente: Elaboración propia.

La Figura 34 presenta la carga de evidencias, disponible únicamente cuando el proyecto está Aprobado o Iniciado; el archivo se restringe a documentos PDF y se acompaña de un nombre descriptivo.

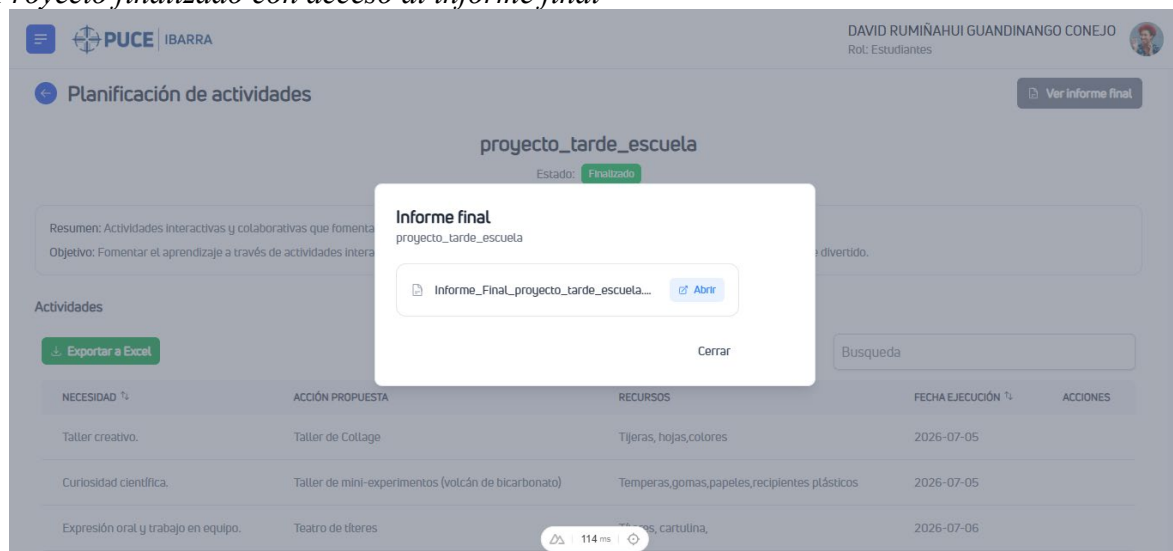
Figura 35
Finalización del proyecto con informe final obligatorio



Fuente: Elaboración propia.

La Figura 35 muestra la ventana de finalización: el sistema advierte que la acción es irreversible y exige adjuntar el informe final en PDF antes de confirmar el cierre del proyecto.

Figura 36
Proyecto finalizado con acceso al informe final



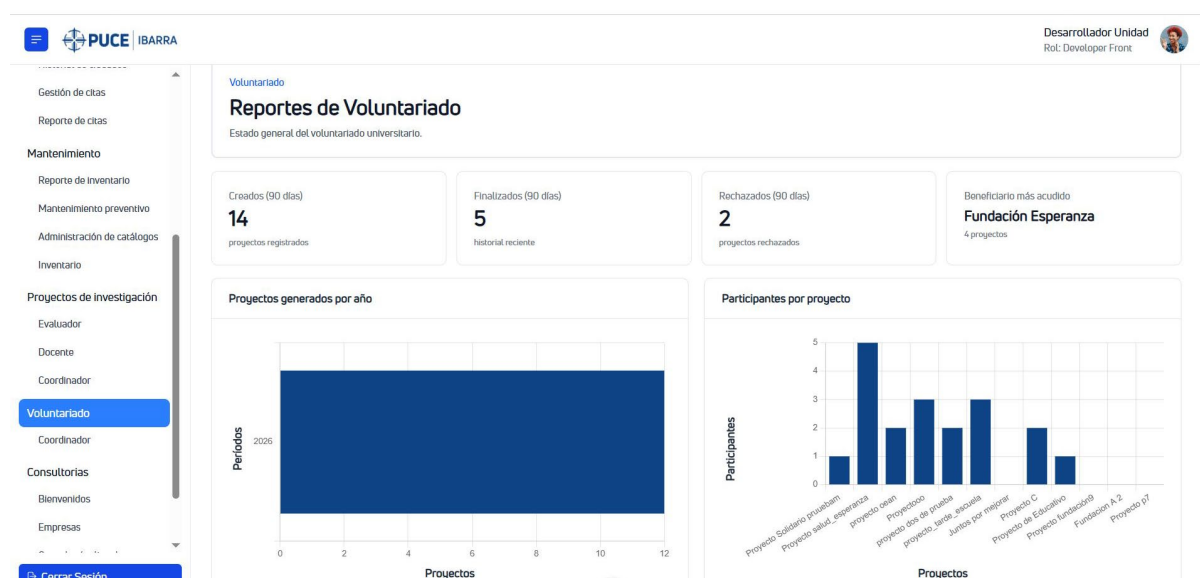
Fuente: Elaboración propia.

La Figura 36 presenta el panel del líder con el proyecto en estado Finalizado y el acceso Ver informe, que permite consultar el documento de cierre; esta consulta también está disponible para el coordinador.

3.1.3. Módulo de reportes del voluntariado

El módulo de reportes materializa la visualización de información centralizada del voluntariado y está disponible para ambos roles, de modo que tanto el coordinador como el líder de proyecto pueden consultar el estado general de la gestión. Este módulo consolida en un único tablero los datos operativos generados a lo largo del ciclo de vida de los proyectos y los presenta mediante indicadores clave (KPI) y gráficos estadísticos, con el fin de apoyar la toma de decisiones. Para preservar la coherencia de la información, los reportes excluyen los proyectos en estado Rechazado y los períodos se determinan de forma dinámica a partir de los datos existentes.

Figura 37
Tablero de reportes con indicadores clave (KPI)

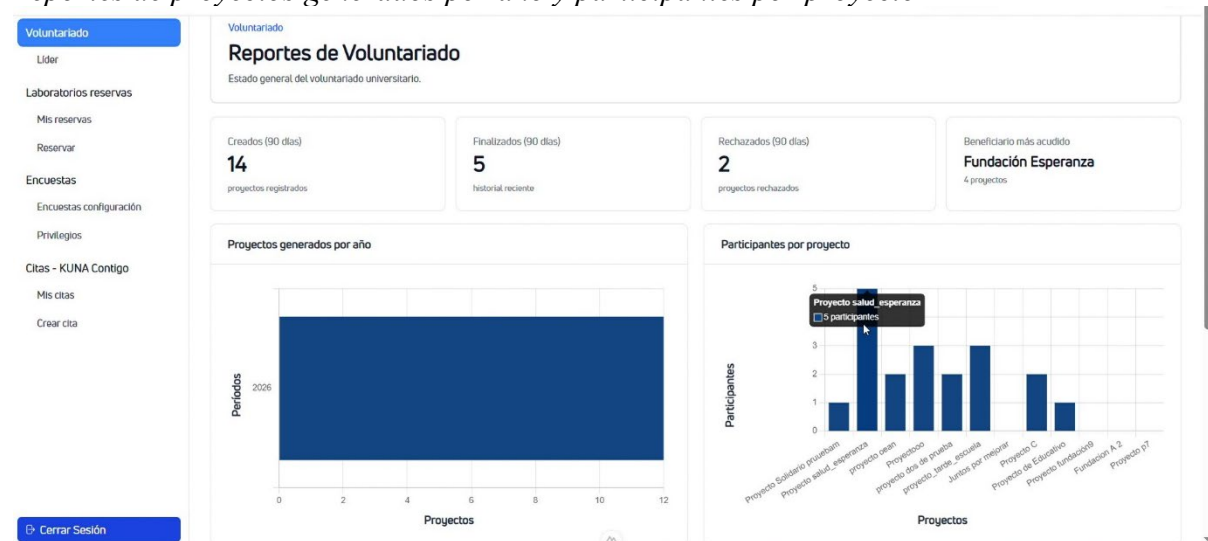


Fuente: Elaboración propia.

La Figura 37 presenta el encabezado del tablero de reportes junto a cuatro indicadores clave que resumen la actividad reciente del voluntariado. Los tres primeros contabilizan, sobre los últimos noventa días, los proyectos registrados, finalizados y rechazados; el cuarto identifica al beneficiario con mayor número de proyectos en el histórico completo. Estos indicadores ofrecen una lectura inmediata del volumen y del desenlace de los proyectos gestionados.

Figura 38

Reportes de proyectos generados por año y participantes por proyecto

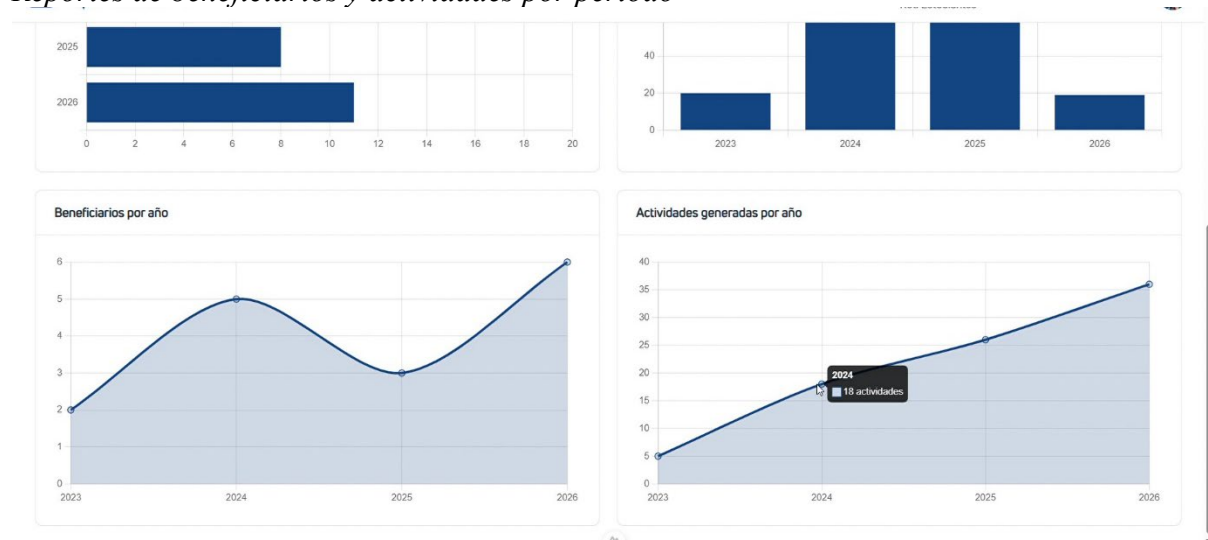


Fuente: Elaboración propia.

La Figura 38 muestra dos reportes representados mediante gráficos de barras. El primero, «Proyectos generados por año», contabiliza los proyectos según su año de creación y permite observar la evolución de la gestión a lo largo del tiempo. El segundo, «Participantes por proyecto», presenta el número de participantes presentes en cada proyecto, lo que facilita comparar la magnitud de la convocatoria entre las distintas iniciativas.

Figura 39

Reportes de beneficiarios y actividades por período



Fuente: Elaboración propia.

La Figura 39 presenta dos reportes de tendencia mediante gráficos de línea. El primero, «Beneficiarios por año», cuantifica los beneficiarios distintos atendidos en cada período, evitando duplicidades cuando una misma entidad participa en varios proyectos. El segundo, «Actividades generadas por periodo», acumula el número de actividades planificadas por año, ofreciendo una visión del esfuerzo operativo desplegado en el voluntariado.

3.2. Validación del sistema (pruebas funcionales)

La validación del sistema se realizó mediante la ejecución manual de las pruebas funcionales definidas en el plan de pruebas del capítulo anterior, utilizando la plantilla de registro de la Tabla 7 y los casos descritos en las Tablas 8 a 15. Las pruebas se ejecutaron al cierre de cada Sprint sobre las funcionalidades terminadas, verificando tanto los escenarios esperados como los casos excepcionales de cada historia de usuario.

Las Tablas 16 a 23 muestran las pruebas de aceptación resueltas, con el estado final de cada escenario ejecutado. Estas tablas permiten validar el estado del producto, pues evidencian el cumplimiento de los criterios de aceptación definidos para cada historia de usuario antes de dar por concluida la iteración correspondiente.

Tabla 16
Prueba unitaria: inicio de sesión

Prueba de Aceptación Nro. 01				
<i>Nombre:</i>	Inicio de sesión del usuario			
<i>Descripción:</i>	El sistema debe permitir el acceso solo a usuarios autorizados, garantizando la seguridad y la asignación de roles.			
<i>Escenario</i>	Usuario	Contraseña	Resultado Esperado	Estado
<i>Credenciales válidas</i>	Usuario institucional registrado	Correcta	El sistema valida las credenciales y redirige al panel correspondiente al rol del usuario.	Aprobado
<i>Credenciales incorrectas</i>	Usuario institucional registrado	Incorrecta	El sistema deniega el acceso y muestra un mensaje de credenciales inválidas.	Aprobado
<i>Usuario no habilitado</i>	Usuario sin registro previo	Cualquiera	El sistema concede el acceso únicamente a usuarios registrados y habilitados.	Aprobado
<i>Rol no habilitado</i>	Usuario con rol sin permisos del sistema	Correcta	El sistema no expone en el menú las rutas del sistema de voluntariado.	Aprobado
<i>Observaciones:</i>	El acceso y las opciones de menú disponibles dependen de los roles configurados en el backend institucional.			

Fuente: Elaboración propia.

Tabla 17

Prueba unitaria: gestión de proyectos de voluntariado

Prueba de Aceptación Nro. 02						
Nombre:		Gestión de proyectos de voluntariado				
Descripción:		El usuario coordinador debe poder registrar, editar y actualizar el estado de los proyectos de voluntariado.				
Escenario		Campo		Entrada	Resultado Esperado	Estado
Registrar proyecto	un	Formulario de creación	de	Completar nombre, descripción, área y beneficiario	El proyecto se registra y aparece en el listado con el estado Creado. La información se actualiza y se refleja en la tabla; la edición solo se habilita en estado Creado.	Aprobado
Modificar proyecto	un	Botón “Editar proyecto”		Actualizar los campos y guardar	El estado se actualiza y se refleja en la columna de estado y en los chips de filtro.	Aprobado
Cambiar el estado		Acciones de la fila		Ejecutar la acción y confirmar		Aprobado
Observaciones:						

Fuente: Elaboración propia.

Tabla 18

Prueba unitaria: solicitud de ejecución del proyecto

Prueba de Aceptación Nro. 03					
Nombre:	Solicitud de ejecución del proyecto				
Descripción:	El líder debe poder enviar la planificación a revisión del coordinador y reenviarla después de un rechazo.				
Escenario		Campo	Entrada	Resultado Esperado	Estado
Enviar a revisión		Botón “Enviar a revisión”	Completar la planificación y confirmar el envío	El proyecto pasa a Enviado a revisión y aparece pendiente en el panel del coordinador.	Aprobado
Reenviar tras un rechazo	un	Botón “Reenviar”	Corregir la planificación y confirmar	El proyecto pasa a Reenviado y vuelve a la bandeja de revisión del coordinador. La opción permanece deshabilitada fuera de los estados Asignado responsable o Rechazado.	Aprobado
Estado habilitado	no	Validación de estado	Intentar enviar en un estado distinto		Aprobado
Observaciones:					

Fuente: Elaboración propia.

Tabla 19*Prueba unitaria: registro de participantes voluntarios*

Prueba de Aceptación Nro. 04					
Nombre:		Registro de participantes en el proyecto			
Descripción:		El líder debe poder registrar, consultar y eliminar a los participantes de sus proyectos asignados.			
Escenario		Campo	Entrada	Resultado Esperado	Estado
Registrar un participante		Formulario registro	de Seleccionar el tipo de voluntario y completar los datos	El participante se registra en el listado de presentes; los perfiles institucionales se autocompletan.	Aprobado
Identificación duplicada		Validación duplicados	de Ingresar la identificación de un participante ya presente	El sistema muestra el mensaje de duplicado e impide el registro.	Aprobado
Eliminación de un participante		Acción eliminación	de Registrar el motivo de eliminación y confirmar	El participante se elimina y pasa al historial de eliminados con su observación.	Aprobado
Observaciones:					

*Fuente: Elaboración propia.***Tabla 20***Prueba unitaria: planificación de actividades del proyecto*

Prueba de Aceptación Nro. 05						
Nombre:		Planificación de actividades del proyecto				
Descripción:		El líder debe poder registrar, editar y desactivar las actividades planificadas del proyecto.				
Escenario		Campo	Entrada		Resultado Esperado	Estado
Registrar actividad	una	Formulario actividad	de	Completar necesidad, acción, recursos y fecha	La actividad se registra asociada al proyecto y actualiza el contador del listado.	Aprobado
Modificar actividad	una	Botón “Editar”		Actualizar información guardar	La actividad se actualiza; la edición solo se habilita mientras la planificación es editable.	Aprobado
Eliminar actividad	una	Interruptor estado	de	Confirmar eliminación	La actividad se elimina del listado y se	Aprobado

	conserva el registro histórico.
Observaciones:	
<i>Fuente: Elaboración propia.</i>	

Tabla 21

Prueba unitaria: autorización del cambio de estado

Prueba de Aceptación Nro. 06					
Nombre:	Autorización del cambio de estado por el coordinador				
Descripción:	El coordinador debe poder aprobar o rechazar la planificación enviada por el líder, registrando el motivo del rechazo.				
Escenario	Campo	Entrada	Resultado Esperado	Estado	
Aprobar proyecto	un Botón “Aprobar”	Revisar la planificación y confirmar decisión	El proyecto pasa a Aprobado y habilita al líder el registro de evidencias.	Aprobado	
Rechazar motivo	con Botón “Rechazar”	Ingresar la observación y confirmar	El proyecto pasa a Rechazado y el motivo queda visible para el líder.	Aprobado	
Reenvío corrección	tras Flujo de reenvío	El líder corrige y reenvía la planificación	El proyecto pasa a Reenviado y se lista como pendiente para el coordinador.	Aprobado	
Observaciones:	La decisión se ejecuta desde la revisión de la planificación, lo que garantiza que el coordinador consulte el plan antes de aprobar o rechazar.				

Fuente: Elaboración propia.

Tabla 22

Prueba unitaria: registro de evidencias de ejecución

Prueba de Aceptación Nro. 07					
Nombre:	Registro de actividades realizadas y evidencias				
Descripción:	El líder debe poder cargar evidencias de la ejecución del proyecto en formato PDF, visibles para el coordinador.				
Escenario	Campo	Entrada	Resultado Esperado	Estado	
Subir evidencia	una Carga de archivo	Adjuntar un documento PDF su descripción	La evidencia se registra y queda visible para el líder y el coordinador.	Aprobado	
Formato permitido	no Validación de archivo	Adjuntar un archivo distinto de PDF	El sistema rechaza el archivo e indica que solo se admite el formato PDF.	Aprobado	

Estado permitido	no	Validación estado	de	Intentar fuera Aprobado Inicializado	cargar de o	La opción de subir evidencias permanece deshabilitada.	Aprobado
Observaciones:							
Fuente: Elaboración propia.							

Tabla 23

Prueba unitaria: cierre del proyecto e informe final

Prueba de Aceptación Nro. 08							
Nombre:		Cierre del proyecto e informe de ejecución					
Descripción:		El líder debe poder finalizar el proyecto adjuntando el informe final en PDF, quedando disponible para consulta.					
Escenario		Campo		Entrada		Resultado Esperado	Estado
Finalizar informe	con	Botón “Finalizar proyecto”		Adjuntar el informe final en PDF y confirmar		El proyecto pasa a Finalizado y el informe queda almacenado.	Aprobado
Finalizar informe	sin	Validación informe	de	Confirmar adjuntar el documento	sin el	El sistema impide la acción e indica que el informe es obligatorio.	Aprobado
Consultar informe	el	Botón informe	“Ver	Abrir el informe de un proyecto finalizado		El informe se visualiza tanto para el líder como para el coordinador.	Aprobado
Observaciones:							
Fuente: Elaboración propia.							

Tabla 24

Prueba unitaria: reportes del voluntariado

Prueba de Aceptación Nro. 09							
Nombre:		Reportes e indicadores del voluntariado					
Descripción:		El sistema debe presentar a ambos roles un tablero de reportes con indicadores y gráficos consistentes con los datos registrados.					
Escenario		Campo		Entrada		Resultado Esperado	Estado
Cálculo indicadores	de	Tarjetas KPI		Abrir el tablero de reportes		Los indicadores muestran los conteos correctos de proyectos y el beneficiario más acudido.	Aprobado
Visualización gráficos	de	Gráficos estadísticos		Consultar los reportes por año y por proyecto		Los gráficos cargan y reflejan los datos por período y por proyecto.	Aprobado

Exclusión de Filtro de datos rechazados	Registrar un proyecto en estado Rechazado	El proyecto rechazado no se contabiliza en los reportes ni en los gráficos.	Aprobado
Observaciones:			

Fuente: Elaboración propia.

3.2.1. Resumen general de resultados

La Tabla 25 consolida los resultados de la validación funcional, detallando por cada historia de usuario el número de escenarios ejecutados y su estado final.

Tabla 25
Resumen de resultados de la validación funcional

Historia de usuario	Escenarios ejecutados	Aprobados	Observaciones
H1. Autenticación y control de roles	4	4	Ninguna
H2. Creación y asignación de proyecto de voluntariado	3	3	Ninguna
H3. Registro de participantes en el proyecto	3	3	Ninguna
H4. Planificación de actividades del proyecto	3	3	Ninguna
H5. Solicitud de ejecución del proyecto	3	3	Ninguna
H6. Autorización del cambio de estado por el coordinador	3	3	Ninguna
H7. Registro de actividades realizadas y evidencias	3	3	Ninguna
H8. Cierre del proyecto e informe de ejecución	3	3	Ninguna
H9. Visualización de reportes e indicadores	3	3	Ninguna
Total	28	28	—

Fuente: Elaboración propia.

En total se ejecutaron 25 escenarios de prueba correspondientes a las nueve historias de usuario implementadas, todos con estado Aprobado. La historia de usuario 9, relativa a la visualización

de información centralizada, se validó mediante la comprobación de los indicadores y los gráficos del módulo de reportes, confirmando la consistencia de la información presentada.

3.2.2. Análisis de hallazgos y observaciones

Durante la ejecución de las pruebas se identificaron y corrigieron oportunamente ajustes menores de la interfaz, entre ellos la unificación de los colores de estado entre los chips de filtro y la columna de estado de la tabla de proyectos, así como la visibilidad condicionada de los botones de acción según el estado de cada fila, aspectos que inicialmente presentaban inconsistencias visuales.

Asimismo, se detectó un comportamiento de actualización en las tablas al editar registros, derivado del componente de tabla compartido bajo el concepto de Nuxt UI; el hallazgo se resolvió forzando la recarga local del componente, sin modificar los componentes reutilizados. De igual forma, se incorporó una validación en vivo del estado del proyecto antes de ejecutar las acciones del líder, con el fin de prevenir conflictos cuando el coordinador cambia el estado mientras la pantalla permanece abierta.

Estos hallazgos evidencian el valor de la validación al cierre de cada Sprint prevista en la metodología, pues permitió detectar y corregir los comportamientos no deseados antes de avanzar a la siguiente iteración.

3.3. Discusión de resultados

Los resultados obtenidos guardan correspondencia con el propósito planteado: el sistema centraliza la información del proceso de voluntariado y organiza el ciclo de vida de los proyectos mediante una máquina de estados que formaliza el flujo entre el coordinador y el líder, desde la creación y asignación hasta la aprobación, la ejecución con evidencias y el cierre con informe final. Con ello se reemplaza el manejo manual y disperso de la información identificado en el diagnóstico inicial. A continuación, estos hallazgos se contrastan con los trabajos analizados en el estado del arte y se interpretan a la luz del problema abordado.

3.3.1. Comparación con trabajos previos

Los resultados de esta investigación coinciden con los de Fraguío Sánchez (2023), quien evidenció que una plataforma web permite centralizar la información del voluntariado y generar indicadores de trazabilidad. No obstante, existen diferencias sustanciales en la solución propuesta. Mientras dicho autor empleó Java en el backend y React en el frontend, el presente trabajo se construyó sobre un ecosistema unificado en TypeScript, con NestJS y Nuxt, lo que reduce la fragmentación tecnológica y facilita el mantenimiento por parte de un mismo equipo. Ambas soluciones coinciden en el uso de PostgreSQL y en la documentación de los servicios mediante Swagger, lo que confirma la pertinencia de estas decisiones técnicas para este tipo de sistemas.

En relación con Aragón et al. (2022), quienes diseñaron un sistema para TECHO El Salvador bajo un modelo de ciclo de vida en cascada, la diferencia principal radica en el alcance alcanzado. Dichos autores reconocieron como limitación que su propuesta se concentró en la gestión operativa, sin integrar la generación de reportes estadísticos automáticos sobre la situación de los proyectos. El presente trabajo aborda precisamente esa brecha: el módulo de reportes implementado genera indicadores y gráficos de forma automática a partir de los datos operativos, sin requerir consolidación manual. Asimismo, el empleo de Scrum, en lugar de un modelo secuencial, permitió validar incrementos al cierre de cada Sprint y ajustar la solución de manera progresiva.

Respecto a Bobadilla (2024), cuyo sistema para la Universidad Católica de Córdoba constituyó un referente conceptual para el diseño de roles y permisos, se identifica una diferencia relevante en el modelo de operación. En dicha propuesta los voluntarios gestionan su propia asistencia, mientras que en el presente sistema el registro de participantes recae en el líder de proyecto. Esta decisión responde a la dinámica institucional de la Dirección de Identidad y Misión, donde el líder conforma y responde por su equipo de trabajo, lo que garantiza la veracidad de la información registrada. Adicionalmente, el presente trabajo incorpora un elemento no abordado por los autores revisados: la integración con el Directorio Activo institucional mediante LDAP, que evita la creación de credenciales paralelas y aprovecha la infraestructura de autenticación ya existente en la universidad.

3.3.2. Interpretación de los resultados

La efectividad de la máquina de estados implementada se explica por su correspondencia con el problema de trazabilidad detectado en el diagnóstico. Al condicionar cada acción al estado del proyecto, el sistema impide transiciones inválidas —como registrar evidencias en un proyecto que aún no ha sido aprobado— y obliga a que toda decisión quede documentada. De esta manera, el estado del proyecto deja de ser una apreciación subjetiva de quien lo consulta y se convierte en un dato verificable, lo que resuelve la dificultad de seguimiento que motivó esta investigación.

En cuanto al módulo de reportes, su relevancia radica en que transforma los datos operativos en información útil para la gestión. Antes de la implementación, obtener un indicador exigía revisar manualmente los archivos remitidos por cada líder; con el sistema, esa información se consolida de forma automática. Esto no solo reduce el esfuerzo administrativo, sino que habilita una gestión basada en evidencia, alineada con la necesidad institucional de sustentar la toma de decisiones sobre el voluntariado universitario.

Los hallazgos surgidos durante las pruebas también aportan lecciones de diseño. La necesidad de forzar la recarga local del componente de tabla y de incorporar una validación en vivo del estado del proyecto antes de ejecutar las acciones del líder evidencian que, en un sistema con múltiples usuarios concurrentes, la consistencia de la información no puede asumirse desde la interfaz. Estos ajustes refuerzan la importancia de validar el estado en el momento de la acción y no únicamente al cargar la pantalla, criterio que resultó determinante para preservar la integridad del flujo de trabajo.

3.3.3. Limitaciones del estudio

El alcance definido para este trabajo determina un conjunto de limitaciones que deben considerarse al interpretar los resultados. En primer lugar, el sistema no contempla roles externos a la universidad, por lo que los representantes de las entidades beneficiarias no interactúan directamente con la plataforma. En segundo lugar, no se incluyó un módulo para la gestión de solicitudes de recursos —como transporte o insumos—, aspecto que se atiende por fuera del sistema. Tampoco se abordó la integración con otros sistemas institucionales ni el desarrollo de una aplicación móvil.

Estas limitaciones acotan el ámbito de aplicación de la solución, pero no comprometen la validez de los resultados obtenidos, puesto que las funcionalidades evaluadas corresponden

íntegramente al alcance definido y al proceso que actualmente ejecuta la Dirección de Identidad y Misión. Cabe señalar, además, que la validación se realizó en base a la fase de pruebas en cada sprint sobrellevada en la metodología de desarrollo adoptada en un entorno controlado, por lo que los resultados reflejan el comportamiento del sistema bajo condiciones del plan de pruebas de forma incremental y no necesariamente bajo una metodología clásica que se basa en una sección de pruebas al final de todo el masivo y prolongado proyecto.

3.3.4. Consideraciones sobre los aspectos no funcionales

La validación desarrollada se concentró en los aspectos funcionales del sistema, es decir, en verificar que las funcionalidades respondieran a los criterios de aceptación definidos. En consecuencia, los atributos no funcionales abordados en el marco teórico —particularmente la usabilidad y la seguridad— no fueron objeto de una evaluación formal mediante instrumentos como el cuestionario SUS o una evaluación heurística, lo que constituye una limitación del presente estudio.

La verificación del sistema se realizó mediante la ejecución de 25 escenarios de prueba, elaborados a partir de los criterios de aceptación definidos para las historias de usuario y ejecutados siguiendo el marco de pruebas propuesto por Scrum. Este proceso permitió verificar el correcto funcionamiento de las funcionalidades implementadas, evidenciando que las nueve historias de usuario desarrolladas cumplen con los requisitos establecidos y responden a las necesidades identificadas durante el levantamiento de requerimientos.

La validación se centró en los aspectos funcionales del sistema, verificando el cumplimiento de los criterios de aceptación definidos para cada historia de usuario. De manera complementaria, los atributos de usabilidad y seguridad orientaron las decisiones de diseño e implementación durante el desarrollo. En cuanto a la usabilidad, la reutilización de componentes de Nuxt UI permitió mantener una interfaz consistente con el entorno institucional ya conocido por los usuarios, mientras que la organización de los flujos de navegación se adaptó a las tareas específicas de cada rol. Respecto a la seguridad, la autenticación mediante el Directorio Activo institucional (LDAP) eliminó la necesidad de gestionar credenciales adicionales y delegó su administración a la infraestructura de la universidad; además, el control de acceso basado en roles restringió las operaciones e información disponibles para cada tipo de usuario. La evaluación de estos atributos no forma parte del presente trabajo, ya que el sistema se integra como un módulo del sistema administrativo de la universidad. En consecuencia, conforme a las políticas y lineamientos

institucionales, dichas pruebas son responsabilidad de los administradores de sistemas y del personal técnico encargado del soporte y la gestión de la infraestructura tecnológica.

3.3.5. Implicaciones prácticas y organizacionales

Desde una perspectiva práctica, el sistema incide directamente en la eficiencia operativa de la Dirección de Identidad y Misión. Actividades que anteriormente exigían solicitar archivos a cada líder y consolidarlos manualmente —como conocer el estado de un proyecto o construir un indicador— se resuelven ahora mediante una consulta directa en la plataforma. Dado el alcance definido para el presente trabajo, se emplearon pruebas de usabilidad que permitieron validar la eliminación de estas tareas intermedias, dando cumplimiento al ciclo de Scrum en su fase de pruebas.

La adopción del sistema supone, asimismo, ajustes organizacionales. Requiere que la información se registre en la plataforma y no en archivos personales, que la aprobación de las planificaciones se realice por este medio y que los responsables asuman el registro oportuno de participantes y evidencias. En este sentido, el sistema no solo automatiza el proceso existente, sino que lo formaliza, lo cual exige acompañar su implantación con capacitación y con la definición institucional de la plataforma como medio oficial de gestión del voluntariado.

CONCLUSIONES

Una vez terminado el desarrollo de la propuesta para la gestión de los proyectos de voluntariado de la Dirección de Identidad y Misión de la PUCE Ibarra, se llega a determinar las siguientes conclusiones:

- El análisis del proceso actual permitió identificar las principales limitaciones de la gestión manual del voluntariado, evidenciando la dispersión de la información y las dificultades en el seguimiento y control de los proyectos, lo que sustentó la necesidad de una solución tecnológica.
- La definición de los requisitos funcionales y no funcionales, apoyada en una metodología de desarrollo ágil, permitió organizar las necesidades de la institución en historias de usuario y priorizar su construcción de manera iterativa, asegurando que la solución respondiera a las expectativas de los usuarios gestores y operativos.
- El diseño de la arquitectura, orientado a la gestión de roles de usuario, permitió integrar los roles diferenciados de coordinador y líder de proyecto, que operan sobre un ciclo de vida completo de los proyectos, desde su creación hasta su finalización. Los estados intermedios garantizan una aprobación controlada y una revisión continua, lo que reduce las transiciones inválidas y mantiene la coherencia de la información en todo el proceso, asegurando que cada acción sea reversible o quede documentada mediante observaciones. Asimismo, la separación clara de responsabilidades favorece que el sistema sea legible y modificable por futuros desarrolladores, sin riesgo de afectar las funcionalidades existentes.
- La validación mediante pruebas de funcionalidad confirmó el cumplimiento de los requisitos establecidos, verificando que el sistema opera de forma correcta y confiable. De esta manera, la información de los proyectos queda centralizada y resguardada, evitando su pérdida y facilitando su consulta en cualquier momento.
- En conjunto, el sistema constituye una base tecnológica sólida que, al centralizar la información y formalizar los flujos de trabajo, supera las limitaciones identificadas en el diagnóstico inicial. Su módulo de reportes aporta, además, indicadores que sustentan la toma de decisiones, de modo que la solución queda lista para su despliegue institucional y cumple con el objetivo general del trabajo.

RECOMENDACIONES

Con el propósito de asegurar la continuidad del sistema de voluntariado y promover su mejora constante dentro del contexto institucional, se plantean las siguientes recomendaciones:

- Incorporar, como línea de evolución, roles externos a la universidad dentro del sistema, de modo que los representantes de las entidades beneficiarias puedan interactuar directamente con la plataforma. Esto permitiría gestionar de forma digital las autorizaciones y convenios de colaboración, dando mayor trazabilidad a la relación entre la institución y las comunidades atendidas.
- Evaluar la ampliación del sistema con un módulo para la gestión de solicitudes y autorizaciones de recursos asociados a la ejecución de los proyectos, como el transporte de los participantes o los insumos y materiales requeridos para las actividades y eventos de voluntariado. Su incorporación complementaría el flujo actual y ordenaría la logística vinculada a cada iniciativa.
- Brindar capacitaciones periódicas a los coordinadores y líderes que operan el sistema, enfocadas en el uso adecuado de las funcionalidades clave —gestión de proyectos, participantes, planificación y evidencias—, con el fin de aprovechar plenamente la herramienta y reducir incidencias durante su operación.
- Realizar un monitoreo periódico del uso del sistema para identificar oportunidades de mejora en la funcionalidad, la experiencia de usuario y el rendimiento, incorporando la retroalimentación de los coordinadores, líderes y demás actores involucrados en los proyectos de voluntariado.
- Ampliar el módulo de reportes con nuevos indicadores orientados a medir el impacto social del voluntariado, tales como el número de beneficiarios atendidos, las horas de participación o la cobertura por comunidad, de manera que la información consolidada fortalezca la toma de decisiones a nivel institucional.
- Implementar un proceso de análisis trimestral de los indicadores generados por el sistema, con el fin de medir el impacto social de los proyectos de voluntariado y alimentar los informes de autoevaluación institucional.
- Establecer un plan de mantenimiento de la infraestructura tecnológica que contemple la actualización periódica de las dependencias del sistema, la realización de respaldos regulares de la base de datos y la renovación oportuna de los certificados de seguridad, con el propósito de garantizar la continuidad operativa y la protección de la información a largo plazo.

- Mantener actualizada la documentación técnica y funcional del sistema, junto con procesos de mantenimiento preventivo y correctivo, para facilitar futuras modificaciones, la incorporación de nuevos requerimientos o eventuales migraciones tecnológicas.

REFERENCIAS BIBLIOGRÁFICAS

- Ahmad, N. (Agosto de 2024). Patrones arquitectónicos y desafíos en Spring Boot para microservicios: evaluación de estrategias de automatización para el escalado, la monitorización y el despliegue en ecosistemas de software complejos. *International Journal of Information Technologies and Artificial Intelligence*, 8(8). Obtenido de ResearchGate:
https://www.researchgate.net/publication/385686967_Architectural_Patterns_and_Challenges_in_Spring_Boot_for_Microservices_Evaluating_Automation_Strategies_for_Scaling_Monitoring_and_Deployment_in_Complex_Software_Ecosystems
- Al-Ibrahim, M., & Shams, Y. (2014). La realidad de la aplicación de la seguridad en las aplicaciones web en el ámbito académico. *Revista Internacional de Ciencias de la Computación Avanzadas y Aplicaciones (IJACSA)*, 5(10), 1-15.
doi:<https://doi.org/10.14569/IJACSA.2014.051002>
- Aragón, M., Domínguez, E., Fuentes, H., & Henríquez, E. (2022). *Sistema informático para la gestión de voluntarios y administración de prouyectos de TECHO El Salvador*. San Salvador: Repositorio Insitucional de la Universidad de El Salvador.
doi:<https://hdl.handle.net/20.500.14492/18244>
- Arroziqi, A., & Susanty, M. (2026). Uniendo arquitecturas monolíticas y de microservicios: una arquitectura monolítica modular para aplicaciones RESTful escalables con NestJS. *SSRN Universidades Pertamina*. Obtenido de https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6042616
- Asamblea Nacional del Ecuador. (2010). *Ley Orgánica de Educación Superios (LOES)*. Quito: Registro Oficial 298, Suplemento. Obtenido de <https://www.ces.gob.ec/documentos/Normativa/LOES.pdf>
- Barde, K. (2023). Monolitos modulares: una revolución en la arquitectura de software para sistemas de pago eficientes en el sector fintech. *Revista Internacional de Tendencias y Tecnologías Informáticas*, 71(10), 20-27.
doi:<https://doi.org/10.14445/22312803/IJCTT-V71I10P103>
- Blinowski, G., Ojdowska, A., & Przybylek, A. (2022). Arquitectura monolítica frente a arquitectura de microservicios: una evaluación de rendimiento y escalabilidad. *Revistas y publicaciones periódicas IEEEExplore*, 10, 20357-20374. Obtenido de <https://ieeexplore.ieee.org/document/9717259>

- Bobadilla, M. (2024). *Sistema voluntario UCC: sistema informático para la gestión de los voluntariados del VUCC y sus actividades*. Córdoba: Universidad Católica de Córdoba. Obtenido de <http://pa.bibdigital.ucc.edu.ar/4159/>
- Celi, R., Miguel, B., & Mora, A. (2023). *Programación WEB del Frontend al Backend*. Santo Domingo-Ecuador: Editorial Grupo AEA. Obtenido de <https://dialnet.unirioja.es/servlet/libro?codigo=933116>
- Cevallos, A. (5 de Noviembre de 2025). *Ibarra fortalece los valores ignacionos e identes a través del encuentro humanístico institucional*. (P. Ibarra, Editor) Obtenido de <https://www.pucesi.edu.ec/web/noticias/?p=8688>
- Cevallos, A. (13 de Diciembre de 2025). *Proyecto de voluntariado "Servir para transformarnos" Realizado durante los meses de noviembre a diciembre*. doi:<https://www.pucesi.edu.ec/web/noticias/?p=8895>
- Chávarry, K., & Sandoval, J. (2023). Transformación digital en la Gestión Documental de la Institución de Educación Superior Tecnológica: Implementación de un Sistema WEB con Metodología XP. *Revista Qantu Yachay*, 3(2), 1-10. doi:<https://doi.org/10.54942/qantuyachay.v3i2.54>
- Dreiser, T. (2024). A comparative study of black box and white box testing techniques in modern software development. *FRONTIERS IN ENGINEERING AND TECHNOLOGY (FET)*, 5(1), 1-7. Obtenido de https://iaeme.com/MasterAdmin/Journal_uploads/FET/VOLUME_5_ISSUE_1/FET_05_01_001.pdf
- Ellislab/CodeIgniter. (3 de Marzo de 2022). Obtenido de El framework CodeIgniter: <https://packagist.org/packages/ellislab/codeigniter>
- Fraguío, J. (2023). *Herramienta para la gestión y coordinación del voluntariado*. Univesridad de la Coruña. Coruña: Repositorio de la Universidad de la Coruña. doi:<http://hdl.handle.net/2183/34084>
- Gongalves, M. (13 de Octubre de 2021). *¿Qué es Angular y para qué sirve?* Obtenido de <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>
- Gordon Graell, R. (2023). Interacción humano-computador y sus aportes en el desarrollo de la informática aplicada a la educación. *Revista Multidisciplinar de la Universidad de Cienfuegos*, 15(2), 110-119. Obtenido de <https://rus.ucf.edu.cu/index.php/rus/article/view/3611>

- Guntakandla, A. (2025). Arquitectura modular: un enfoque de diseño de sistemas escalable y eficiente para aplicaciones empresariales. *Revista Mundial de Investigación y Reseñas Avanzadas*, 31(1), 3114-3126. doi:<https://doi.org/10.30574/wjarr.2025.26.1.1340>
- Hassan, H., Abdel-Fattah, M., & Mohamed, W. (2024). Migrating from monolith to microservice architectures: a systematic literature review. *Revista Internacional de Ciencias de la Computación Avanzadas y Aplicaciones*, 15(10). doi:<https://doi.org/10.14569/IJACSA.2024.0151013>
- IEEE SA. (9 de Junio de 2005). *Norma IEEE para la documentación de pruebas de software y sistemas*. Obtenido de <https://standards.ieee.org/ieee/829/3787/>
- Jihan, K., & Mahummad, M. (2024). Asana y Trello: una evaluación comparativa de sus capacidades de gestión de proyectos. *JOIV Revista Internacional de Visualización Informática*, 8(1), 207-212. doi:<http://dx.doi.org/10.62527/joiv.8.1.2595>
- Keshari, P., Maurya, P., Kumar, P., & Katiyar, A. (2023). Desarrollo web con ReactJS. 5.^a *Conferencia Internacional de 2023 sobre Avances en Computación, Control de Comunicaciones y Redes (ICAC3N)*. IEEE. doi:<https://doi.org/10.1109/ICAC3N60023.2023.10541743>
- Khan, S., & Khanam, A. (2023). Study on MVC framework for web development in PHP. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 9(4), 414-419. doi: <https://doi.org/10.32628/CSEIT2390450>
- Khan, W., Kumar, T., Zhang, C., Raj, K., Roy, A., & Luo, B. (2023). SQL and NoSQL database software architecture performance analysis and assessments. *Publicado por MDPI*, 7(2). doi:<https://doi.org/10.3390/bdcc7020097>
- Khokhriagkov, I. (26 de Septiembre de 2024). *Performance Benchmarking of Node.js, Rust, Go, and Python in WEB Applications*. Obtenido de NOBUGS 2024: <https://indico.esrf.fr/event/114/contributions/777/>
- Kyriakou, K., & Tselikas, N. (2022). Complementing JavaScript in High-Performance Node.js and Web Applications with RUST and WEBAssembly. *Revista Electronics MDPI*, 11(19), 1-17. doi:<https://doi.org/10.3390/electronics11193217>
- Lando, B., & Hasselbring, W. (2025). Hacia federaciones de módulos independientes de Bundler: habilitando arquitecturas de micro-frontend tipadas. En 2. C. 2025) (Ed.). 2025 IEEE 22nd International Conference in Software Architecture Comparison 36-40. Obtenido de <https://arxiv.org/pdf/2501.18225>
- Leones, W., Macias, L., Pilla, P., & Fernández, E. (2024). Diseño Estratégico de APIs Escalables y Seguras para la Integración de Sistemas y Aplicaciones. *Ciencia Latina*

- Revista Multidisciplinar*, 8(5), 13395-13408.
doi:https://doi.org/10.37811/cl_rcm.v8i5.14794
- Marculescu, B., Zhang, M., & Arcuri, A. (2022). Sobre los fallos encontrados en las API REST mediante la generación de pruebas automatizadas. *Revista ACM Digital Library*, 41, 1-43. doi:<https://doi.org/10.1145/3491038>
- Marín, J. (2025). *Desarrollo de una plataforma web para la oferta de servicios especializados brindados por la comunidad estudiantil de la ULEAM*. Manta: Repositorio Intitucional ULEAM. Obtenido de <https://repositorio.uleam.edu.ec/handle/123456789/10548>
- Modi, S., Taher, H., & Mahmud, H. (2021). Una herramienta para automatizar la evaluación de diagramas UML de estudiantes. *Revista Académica de la Universidad de Nawroz*, 10(2), 189-198. doi:[10.25007/ajnu.v10n2a1035](https://doi.org/10.25007/ajnu.v10n2a1035)
- MySliwiec. (2017). *A progressive node.js framework*. doi:<https://nestjs.com/>
- Nadipalli, R. (2024). Progressive web apps with angular: enhancing cloud access on mobile devices. *Journal of Artificial Intelligence & Cloud Computing*, 3(4), 1-5. doi:[https://doi.org/10.47363/JAICC/2024\(3\)486](https://doi.org/10.47363/JAICC/2024(3)486)
- Necula, S. (2024). *Explorando la arquitectura Modelo-Vista-Controlador (MVC): Un análisis exhaustivo de las aplicaciones tecnológicas y de mercado*. Preprints.org.10. doi:<https://doi.org/10.20944/preprints202404.1860.v1>
- Ntoa, S. (2024). Evaluación de la usabilidad y la experiencia del usuario en entornos inteligentes: una revisión y reevaluación. *Revista Internacional de Interacción Humano-Computadora*, 41(5), 2829-2858. doi:<https://doi.org/10.1080/10447318.2024.2394724>
- Núñez, S. (2021). *Implementación de un modelo de desarrollo de software seguro*. Tesis doctoral, Universidad de Extremadura. Obtenido de <http://hdl.handle.net/10662/12316>
- Pantaleo, G., & Rianaudo, L. (2015). *Ingeniería de software*. Buenos Aires - Argentina: Alfaomega Grupo Editor Argentino. Obtenido de https://api.pageplace.de/preview/DT0400.9786076222379_A43569154/preview-9786076222379_A43569154.pdf
- Patel, R. (2025). *What's Scrum?*. Space-O Technologies. doi:<https://www.spaceo.ca/glossary/software-terms/what-is-scrum/>
- Pham, A. (2020). *Developing back-end of a web applications with NestJS Framework: Case: Integrity Oy's student management system*. University of Applied Sciences Ltd. doi:[https://www.theseus.fi/bitstream/handle/10024/353200/Pham_Duc.pdf?sequence=](https://www.theseus.fi/bitstream/handle/10024/353200/Pham_Duc.pdf?sequence=2)

- Pirdaus, D., & Hidayana, R. (2024). Análisis y pruebas de caja negra y caja blanca en una aplicación web basada en listas de tareas. *Revista Internacional de Matemáticas, Estadística e Informática*, 2(2), 68-75. Obtenido de https://www.researchgate.net/publication/380358357_Analysis_Testing_Black_Box_and_White_Box_on_Application_To-Do_List_Based_Web
- Pivotal. (2024). *Pivotal Software*. Obtenido de Biography: <https://openexpoeurope.com/es/company/pivotal/>
- PostgreSQL Global Development Group. (16 de Julio de 2026). *About PostgreSQL*. Obtenido de <https://www.postgresql.org/about/>
- Pressman, R., & Maxim, B. (2014). *Ingeniería de software: Un enfoque práctico*, (8va Edición ed.). McGraw-Hill. Obtenido de https://www.researchgate.net/publication/305000760_Software_Engineering_A_Practitioner's_Approach_8th_Ed
- PUCEI. (25 de Noviembre de 2025). *Fortalece los valores ignacianos e identes a través del encuentro humanístico institucional*. Obtenido de <https://www.pucesi.edu.ec/web/noticias/?p=8688>
- Ramesh, A., Parthiban, A., Jerald, F., Sujitha, M., & Nisha, M. (2025). Diseño e implementación de un sistema de gestión de proyectos y finanzas utilizando PHP y MYSQL. *Revista Internacional de Ciencia y Tecnología*, 16(2), 1-19. doi:<https://doi.org/10.71097/IJSAT.v16.i2.3962>
- Sancho-Nuñez, J. (2021). *Publicación: Implementación de un modelo de desarrollo de software seguro*. Universidad de Extremadura. Repositorio Institucional. Obtenido de <https://dehesa.unex.es/entities/publication/fcacce4d-1845-4570-9e37-56b7c3f4233d>
- Schwaber, K., & Sutherland, J. (Noviembre de 2020). *La guía de Scrum: La guía definitiva de Scrum: Las reglas del juego*. Scrum Guides. Obtenido de <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Stevens, E. (6 de Diciembre de 2024). *¿Qué es el diseño de interfaz de usuario? Una guía completa para 2025*. doi:<https://uxdesigninstitute.com/blog/what-is-ui-design/>
- Su, R., & Li, X. (2024). Monolito modular: ¿Es esta la tendencia en la arquitectura de software? *IEEE/ACM International Workshop New Trends in Software Architectur*, 10-13. doi:<https://doi.org/10.1145/3643657.3643911>

- Suescum, C., Suescum, C., Suescum, C., & Coelho, C. (2025). Transformación digital y ODS: unva revisión sistemática de sinergias y riesgos. *Sapiens International Multidisciplinary Journal*, 2(3), 1-18. doi:<https://doi.org/10.71068/regfc279>
- Tu, Z. (2023). Investigación sobre la aplicación de la arquitectura por capas en el desarrollo de software informático. *Journal of Computing and Electronic Information Managgement*, 11(3). doi:<https://doi.org/10.54097/jceim.v11i3.08>
- Vue.js. ((s.f.)). *Renderizado del lado del servidor*. doi:<https://vuejs.org/guide/scaling-up/ssr.html>
- White, J., Hays, S., Fu, Q., Spencer-Smith, J., & Schmidt, D. (2024). Patrones de indicaciones de ChatGPT para mejorar la calidad del código, la refactorización, la obtención de requisitos y el diseño de software. Generative AI for effective software development. doi:https://doi.org/10.1007/978-3-031-55642-5_4
- Yelin, L. (2025). Gestión de voluntarios en organizaciones sin fines de lucro: Experiencia de la Fundación Huellas en Medellín, Colombia. *Revista Administrative Sciences MDPI*, 15(3). doi:<https://doi.org/10.3390/admsci15030077>

ANEXOS

Anexo 1. Guion de entrevista

Guion de entrevista para el levantamiento de requerimientos

Objetivo: Recopilar información sobre el proceso actual de gestión de proyectos de voluntariado, con el fin de identificar necesidades, limitaciones y requerimientos para el desarrollo de un sistema web.

- ¿Cómo se realiza actualmente el registro o creación de proyectos de voluntariado?
- ¿Qué información se necesita registrar para crear un proyecto de voluntariado?
- ¿Quiénes intervienen en el proceso de gestión de los proyectos de voluntariado?
- ¿Qué roles y qué permisos está previsto tener cada uno?
- ¿Cómo se asigna actualmente un líder responsable a cada proyecto?
- ¿Cómo se inscriben o postulan los voluntarios a un proyecto?
- ¿Qué datos debe proporcionar el voluntario al momento de postularse?
- ¿Cómo se lleva el control de los voluntarios inscritos en cada proyecto?
- ¿Qué dificultades se presentan actualmente en el seguimiento de los participantes?
- ¿Cómo se actualiza el estado de un proyecto, por ejemplo: pendiente, en ejecución, finalizado o cancelado?
- ¿Qué información debería visualizar el coordinador en un tablero o panel de seguimiento?
- ¿Qué filtros serían útiles para consultar la información de los participantes?
- ¿Es necesario que el sistema permita dar de baja a voluntarios inscritos? ¿En qué casos?
- ¿Qué tipo de reportes o información consolidada sería importante obtener desde el sistema?
- ¿Cómo debería realizarse la evaluación del proyecto por parte del voluntario?
- ¿Qué beneficios espera obtener con la implementación de un sistema web para la gestión de voluntariado?
- ¿Se manejan documentos que autoricen el inicio o planificación de un proyecto?
- ¿Se manejan documentos que autoricen el cierre de determinado proyecto?
- ¿Se maneja algún documento para solicitar adquisiciones o recursos para ejecución de un proyecto?
- ¿Qué aspectos considera indispensables para que el sistema sea fácil de usar?
- ¿Existe alguna restricción o consideración importante que deba tomarse en cuenta durante el desarrollo del sistema?

Anexo 2. Ficha de Planificación de Actividades de voluntariado



FASE : ACCIÓN

Instrumento 1: Formato para planificación de la acción

ÁREA DE VOLUNTARIADO: ... PLAN DE ACCIÓN					
1. Resumen del diagnóstico a modo de línea base					
2. Objetivo específico					
3. Planificación de actividades					
Nº	Necesidad detectada	Acción propuesta	Responsable	Recursos	Fecha de ejecución
4. Evidencias o productos esperados					

Instrumento 2: Registro del accionar (opcional)

- a) Datos informativos
- b) Desarrollo de la actividad o actividades
- c) Resultados obtenidos y dificultades encontradas
- d) Reflexión personal y grupal
- e) Conclusiones
- f) Registro video fotográfico

Nota: el docente está en libertad de usar esta estructura o a su vez mejorarla. El medio para su presentación queda al buen criterio de cada docente.

Dirección: Av. Jorge Guzmán Rueda y Av. Aurelio Espinosa Pólit. Ciudadela "La Victoria".
Teléf: (593-6) 2615 500 / 2615 453 Ext. 1000 Cel. 099 236 27 13 / 098 138 3498
Ibarra - Ecuador / www.pucesi.edu.ec



Anexo 3. Carta de recepción de la propuesta tecnológica.



**Pontificia Universidad
Católica del Ecuador**
Seréis mis testigos

IBARRA

DIRECCIÓN DE INVESTIGACIÓN,
VINCULACIÓN Y EMPRENDIMIENTO

Ibarra, 6 de julio de 2026

CARTA DE ACEPTACIÓN

Por medio de la presente, la Pontificia Universidad Católica del Ecuador Sede Ibarra (PUCE-I), a través del Área de Desarrollo de Software de la Oficina de TI, tiene el agrado de informar que el Sr. **David Rumiñahui Guandinango Conejo**, estudiante de la carrera de Ingeniería en Tecnologías de la Información, realizó la entrega del proyecto de titulación titulado:

“SISTEMA WEB PARA LA GESTIÓN DE LOS PROYECTOS DE VOLUNTARIADO DE LA DIRECCIÓN DE IDENTIDAD Y MISIÓN DE LA PUCE IBARRA”

El proyecto fue desarrollado con las políticas y lineamientos establecidos por el Área de Desarrollo de Software de la Oficina de TI de la PUCE Ibarra, el cual se encuentra alojado en los servidores de producción.

Se deja constancia de que el proyecto fue entregado correctamente y cumple con los requerimientos establecidos por el Área.

Atentamente,

Mgs. José Luis Ibarra Estévez

**DIRECTOR - ÁREA DE DESARROLLO DE SOFTWARE DE LA OFICINA DE TI
PUCE-I**