

REPÚBLICA DEL ECUADOR



**DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE
INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICHWA SAN
CARLOS**

Trabajo de Integración Curricular presentado como requisito parcial para optar por el
título de Tecnólogo Superior en Desarrollo de Software

AUTOR: Tapuy Andy Medardo Pedro

TUTOR: Ing. Gonzalo Guanipatín

Tena - Ecuador

2025-IIS

APROBACIÓN DEL TUTOR

ING.GONZALO GUANIPATÍN

DOCENTE DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA

CERTIFICA:

En calidad de Tutor del Proyecto Integrador denominado: DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICIIWA SAN CARLOS, de autoría del señor Tapuy Andy Medardo Pedro, con CC. 1500876774, estudiante de la Carrera de Desarrollo de Software del Instituto Superior Tecnológico Tena, CERTIFICO que se ha realizado la revisión prolija del Trabajo antes citado, el cual cumple con los requisitos de fondo y de forma que exigen los respectivos reglamentos e instituciones.

Tena, 05 de enero de 2026


Ing. zalo Guanipatín
Gon DEL TIC
TUTOR

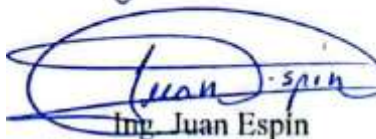
CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR

Tena, 23 de enero de 2026

Los Miembros del Tribunal de Grado abajo firmantes certificamos que el Trabajo de Titulación denominado: DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICIIWA SAN CARLOS, presentado por MEDARDO PEDRO TAPUY ANDY, con CC: 1500876774, estudiante de la Carrera de Desarrollo de Software del Instituto Superior Tecnológico Tena, ha sido corregido y revisado; por lo que autorizamos su presentación.

Atentamente,


Ing. Italo Lara
PRESIDENTE DEL TRIBUNAL


Ing. Juan Espin

MIEMBRO DEL TRIBUNAL (Ü(fl)|

Ing. arco Guanipatín
MIEMBRO DEL TRIBUNAL

AUTORÍA

AUTORÍA

Yo, Medardo Pedro Tapuy Andy, con CC: 1500876774, declaro ser autor del presente Trabajo de Titulación denominado: **DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICHWA SAN CARLOS**. Absuelvo expresamente al Instituto Superior Tecnológico Tena y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido del mismo.

Adicionalmente, acepto y autorizo al Instituto Superior Tecnológico Tena la publicación de mi Trabajo de Titulación en el repositorio institucional-biblioteca virtual.

AUTOR:



MEDARDO PEDRO TAPUY ANDY
CÉDULA: 1500876774

MEDARDO PEDRO TAPUY ANDY
CÉDULA: 1500876774

FECHA: Tena, 05 de enero de 2026

CARTA DE AUTORIZACIÓN POR PARTE DEL AUTOR

Yo, **MEDARDO PEDRO TAPUY ANDY**, declaro ser autor del Trabajo de Titulación titulado: **DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICHWA SAN CARLOS**, como requisito para la obtención del Título de: **TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE**.

Autorizo al Sistema Bibliotecario del Instituto Superior Tecnológico Tena para que, con fines académicos, muestre al mundo la producción intelectual del Instituto a través de la visualización de su contenido que constará en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior con las cuales tenga convenio el Instituto. El Instituto Superior Tecnológico Tena no se responsabiliza por el plagio o copia del presente trabajo que realice un tercero.

Para constancia de esta autorización, en la ciudad de Tena, 05 de enero de 2026, firma el autor.

AUTOR: Medardo Pedro Tapuy Andy

CÉDULA: 1500876774

DIRECCIÓN: Puerto Napo, Comunidad Kichwa San Carlos, km 8 Vía Ahuano

CORREO ELECTRÓNICO: medardo.tapuy@est.itstena.edu.ec

CELULAR: 092006703

DATOS COMPLEMENTARIOS

TUTOR: Ing. Gonzalo Guanipatín

TRIBUNAL DEL GRADO:

Ing. Italo Lara

Ing. Juan Espín

Ing. Marco Guanipatín

DEDICATORIA

Dedico este trabajo a mi familia, especialmente a mis padres y a mi esposa e hijos cuyo apoyo incondicional ha sido el pilar fundamental durante todo mi proceso de formación académica.

A la Comunidad Kichwa San Carlos, por confiar en mí para desarrollar una herramienta que contribuya a su crecimiento organizacional y cultural.

A mis profesores y compañeros del Instituto Superior Tecnológico Tena, por su guía y colaboración.

Y a todos aquellos que creen en el poder de la tecnología para transformar realidades sociales.

AGRADECIMIENTO

Expreso mi sincero agradecimiento:

Al Ing. Gonzalo Guanipatín, mi tutor, por su invaluable orientación, paciencia y dedicación durante todo el desarrollo de este proyecto.

A la Comunidad Kichwa San Carlos, en especial a su directiva, por abrirme las puertas, compartir sus necesidades y participar activamente en la validación del sistema.

Al Instituto Superior Tecnológico Tena, por brindarme la formación académica y los recursos necesarios para llevar a cabo este proyecto.

A mis compañeros de carrera, por el intercambio de conocimientos y el apoyo mutuo durante estos años de estudio.

A mi familia, por su amor, comprensión y motivación constante que siempre me han dado fuerzas para seguir adelante y ser quien soy ahora.

ÍNDICE GENERAL

REPÚBLICA DEL ECUADOR	i
APROBACIÓN DEL TUTOR.....	¡Error! Marcador no definido.
CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR.....	ii
AUTORÍA.....	iii
CARTA DE AUTORIZACIÓN POR PARTE DEL AUTOR	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL	vii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xi

A. TEMA.....	12
B. RESUMEN.....	13
ABSTRACT.....	13
C. FUNDAMENTACIÓN DEL TEMA	14
3.1 Necesidad.....	14
3.2 Actualidad.....	15
3.3 Importancia	16
3.4 Presentación del problema profesional a responder.....	16
3.5 Delimitación.....	17
3.5.1 Delimitación Espacial.....	17
3.5.2 Delimitación Temporal.....	17
3.5.3 Delimitación Técnica.....	17
3.5.4 Unidades de Observación	19
3.6 Beneficiarios	19
3.6.1 Beneficiarios Directos	19
3.6.2 Beneficiarios Indirectos	19
D. OBJETIVOS.....	19
4.1 Objetivo General.....	19
4.2 Objetivos Específicos	20
E. ASIGNATURAS INTEGRADORAS	20
F. FUNDAMENTACIÓN TEÓRICA	21
5.1 Sistemas de Información.....	21
5.2 Metodología ágil – SCRUM.....	21
5.3 Lenguajes de programación	22
5.4 Lenguaje de programación Python	22

5.5 Entorno de desarrollo Visual Studio Code.....	22
5.6 Gestor de base de datos.....	23
5.7 MySQL	23
5.8 Framework Flask	24
5.1.9 Software de gestión.....	24
5.10 Sistemas Web.....	25
5.11 Lenguajes y Tecnologías de Programación Web.....	26
5.12 Diseño centrado en el usuario	26
5.2 Marco Legal	28
5.3 Marco Conceptual.....	29
METODOLOGÍA.....	32
6.1 Materiales.....	32
Recursos de Software.....	32
Recursos Humanos	33
Materiales de Campo.....	33
6.2 Ubicación del Área de Estudio	33
6.3 Tipo de Investigación/Estudio	35
6.4 Metodología par Objetivo.....	35
G. RESULTADOS	58
Resultado del análisis del proceso: Objetivo Específico 1	58
Resultado del diseño de la arquitectura: Objetivo Específico 2	66
Resultado del desarrollo del sistema: Objetivo Específico 3.....	67
H. CONCLUSIONES	68
I. RECOMENDACIONES.....	69
J. BIBLIOGRAFÍA	69

K. ANEXOS..... 71

ÍNDICE DE TABLAS

Tabla 1: Asignaturas integradoras.....	21
Tabla 2: Planificación Sprints	44

ÍNDICE DE FIGURAS

Figura 1: Ubicación de la comunidad	35
Figura 2.- Diagrama Entidad-Relación.....	41
Figura 3.- Wireframe de baja fidelidad - Pantalla principal.....	42
Figura 4.- Prototipo de alta fidelidad-Pantalla principal	43
Figura 5.- Cronograma de Sprints	45
Figura 6.- Interfaz de inicio de sesión	48
Figura 7.- Formulario de registro de miembros.....	50
Figura 8.- Interfaz de gestión de directiva.....	52
Figura 9.- Formulario de carga de documentos.....	53
Figura 10.- Formulario de registro de eventos	54
Figura 11.- Ejemplo de reporte generado en PDF.....	56
Figura 12: Módulo de Autenticación	62
Figura 13: Módulo de Gestión de Miembros	63
Figura 14: Módulo de Gestión de Directiva.....	64
Figura 15: Módulo de Gestión de Documentos	65
Figura 16: Módulo de Gestión de Eventos.....	66
Figura 17: Módulo de Reportes.....	67
Figura 18: Tabla principal del proyecto.....	69
Figura 19: Panel de Control	70
Figura 20: Documentos físicos actuales de la comunidad	76

A. TEMA

DESARROLLO DE UN SISTEMA WEB PARA LA ADMINISTRACIÓN DE INFORMACIÓN ORGANIZACIONAL DE LA COMUNIDAD KICHWA SAN CARLOS

B. RESUMEN

El presente Trabajo de Integración Curricular documenta el diseño y desarrollo de una plataforma web orientada a modernizar la gestión administrativa de la Comunidad Kichwa San Carlos, ubicada en la parroquia Puerto Napo. El proyecto surge como respuesta a una problemática operativa crítica: la ausencia de un sistema digital integrado, lo que obligaba al uso de registros físicos vulnerables. Esta situación provocaba ineficiencias en el manejo de datos, riesgos de pérdida de información y demoras excesivas en los procesos de consulta y actualización de los expedientes institucionales. El objetivo central fue implementar una aplicación web responsiva capaz de centralizar los procesos comunitarios. Metodológicamente, se adoptó el marco de trabajo ágil SCRUM para un desarrollo iterativo. La arquitectura técnica se fundamentó en el uso del framework Flask con Python para el backend, SQL como gestor de base de datos y Bootstrap para garantizar una interfaz gráfica intuitiva y adaptable a diversos dispositivos. Los resultados obtenidos presentan una plataforma robusta que integra módulos de administración de miembros, repositorio documental, gestión de eventos y generación de reportes. El sistema incorpora estrictos protocolos de seguridad, autenticación de usuarios y operaciones CRUD completas. Se concluye que la herramienta optimiza drásticamente los tiempos de respuesta administrativa, asegura la integridad de la información y garantiza la preservación digital del acervo histórico organizacional. Este proyecto representa un hito en la modernización de la Comunidad Kichwa San Carlos, fortaleciendo su gobernanza mediante soluciones tecnológicas contemporáneas y estableciendo un modelo referente para la transformación digital de las comunidades indígenas del Ecuador.

Palabras clave: Comunidad Kichwa, gestión administrativa, plataforma web, sistemas de información, transformación digital.

ABSTRACT

This Curricular Integration Work documents the design and development of a web platform aimed at modernizing the administrative management of the Kichwa San Carlos Community, located in the Puerto Napo parish. The project arises as a response

to a critical operational issue: the absence of an integrated digital system, which necessitated the use of vulnerable physical records. This situation caused inefficiencies in data handling, risks of information loss, and excessive delays in the processes of consulting and updating institutional records. The central objective was to implement a responsive web application capable of centralizing community processes. Methodologically, the SCRUM agile framework was adopted for iterative development. The technical architecture was based on the use of the Flask framework with Python for the backend, SQL as the database management system, and Bootstrap to ensure an intuitive and adaptable graphical interface for various devices. The obtained results present a robust platform that integrates modules for member administration, a document repository, event management, and report generation. The system incorporates strict security protocols, user authentication, and complete CRUD operations. It is concluded that the tool drastically optimizes administrative response times, ensures information integrity, and guarantees the digital preservation of the organizational historical heritage. This project represents a milestone in the modernization of the Kichwa San Carlos Community, strengthening its governance through contemporary technological solutions and establishing a reference model for the digital transformation of indigenous communities in Ecuador.

Keywords: Kichwa Community, administrative management, web platform, information systems, digital transformation.

C. FUNDAMENTACIÓN DEL TEMA

3.1 Necesidad

La Comunidad Kichwa San Carlos actualmente carece de un sistema centralizado para la gestión de su información organizacional, lo que genera dificultades en el registro, actualización y consulta de datos importantes de la comunidad. El flujo de la información organizacional, pilar de su identidad y gestión, está fracturado, ya que se encuentra disperso en varios documentos físicos (cuadernos de trabajo).

La realidad operativa muestra que la valiosa información de la comunidad desde los censos de miembros, las actas de asambleas, los registros de directiva, hasta los datos de proyectos y el patrimonio cultural se encuentra peligrosamente dispersa. Gran parte de esta documentación reside en archivos físicos susceptibles al deterioro, la pérdida accidental o simplemente al paso del tiempo. A esto se suma una colección de archivos digitales sin estructura unificada, lo que se traduce en un problema crónico de duplicidad de registros y dificultad para confirmar la veracidad de los datos.

Esta carencia de un sistema centralizado impacta directamente en la eficiencia administrativa. Los miembros de la directiva, en lugar de enfocarse en la planificación y la gestión, dedican tiempo considerable a la búsqueda manual y frustrante de documentos, un cuello de botella que retrasa cualquier iniciativa. El riesgo es doble: pérdida de tiempo valioso y, más grave aún, la posible extinción de datos históricos irremplazables para la memoria y la continuidad cultural de la comunidad.

La necesidad, por tanto, no es solo técnica, sino fundamentalmente organizacional y cultural. La dificultad para acceder a información fidedigna y actualizada dificulta significativamente la toma de decisiones administrativas informadas, paraliza la planificación de actividades comunitarias y pone en peligro la preservación integral del patrimonio de la comunidad Kichwa. Por ello, es fundamental implementar un sistema web que permita centralizar, organizar y gestionar eficientemente toda la información organizacional de manera segura y accesible.

3.2 Actualidad

En la era digital actual, las organizaciones comunitarias están adoptando tecnologías de información para mejorar su gestión administrativa y preservar su identidad cultural. Los sistemas web de gestión se han convertido en herramientas esenciales para comunidades indígenas que buscan modernizar sus procesos sin perder su esencia tradicional.

Como señala la adopción de las Tecnologías de la Información y la Comunicación (TIC) no es solo una moda, sino un imperativo que moderniza instituciones e incrementa la eficacia operativa. A nivel nacional e internacional, existe una tendencia creciente hacia la digitalización de archivos comunitarios,

implementación de bases de datos para registro de miembros, gestión documental electrónica y plataformas colaborativas que facilitan la participación ciudadana.

El desarrollo de este tipo de sistemas responde a la necesidad de optimizar recursos, mejorar la transparencia administrativa, facilitar el acceso a la información y garantizar la continuidad organizacional mediante la preservación digital del conocimiento comunitario, todo esto alineado con los estándares modernos de gestión de personal administrativo.

3.3 Importancia

La implementación de este sistema web es de vital importancia para la Comunidad Kichwa San Carlos por múltiples razones:

1. **Preservación de información:** Garantizará la preservación y protección de la información organizacional mediante respaldos digitales seguros, evitando la pérdida de documentos históricos y datos relevantes.
2. **Optimización de procesos:** Optimizará los procesos administrativos al reducir tiempos de búsqueda y consulta de información, mejorando la eficiencia del personal administrativo.
3. **Transparencia:** Facilitará la transparencia y rendición de cuentas ante los miembros de la comunidad al proporcionar acceso controlado a información relevante.
4. **Fortalecimiento cultural:** Contribuirá al fortalecimiento de la identidad cultural al documentar digitalmente aspectos organizacionales de la comunidad Kichwa.
5. **Escalabilidad:** Servirá como base tecnológica para futuras expansiones funcionales y representará un modelo replicable para otras comunidades indígenas que busquen modernizar su gestión administrativa.

3.4 Presentación del problema profesional a responder

La Comunidad Kichwa San Carlos carece de un sistema digital integrado para la gestión eficiente de su información organizacional, lo que ha ocasionado pérdida de

información, dificultades en la búsqueda de datos y limitaciones en la toma de decisiones administrativas.

- **Campo:** Tecnologías de la Información y Comunicación
- **Área:** Desarrollo de Sistemas Web
- **Aspecto:** Gestión de Información Organizacional
- **Sector:** Desarrollo de Software
- **Línea de investigación:** Tecnologías de la información y comunicación.

3.5 Delimitación

3.5.1 Delimitación Espacial

El Trabajo de Integración Curricular se realizará en el Instituto Superior Tecnológico Tena, ubicado en la vía Tena-Archidona, km 1.5, pero el sistema será implementado para la Comunidad Kichwa San Carlos, ubicada en la Parroquia Puerto Napo, km 8 vía Ahuano, Provincia de Napo, Ecuador.

3.5.2 Delimitación Temporal

Este proyecto se lo efectuó en el Período Académico 2025-IIS.

3.5.3 Delimitación Técnica

El sistema web presenta un mecanismo de autenticación para el ingreso de usuarios con roles diferenciados de tipo administrador y usuario estándar.

El menú principal del sistema contiene los siguientes módulos:

- **Inicio**, donde puede: visualizar el dashboard principal con estadísticas generales y accesos rápidos.
- **Visión y Misión**, donde puede: consultar y editar la información de la comunidad.
- **Socios**, donde puede: ver proyectos, documentos y su información personal
- **Directiva**, donde puede: acceso básico al sistema.
- **Secretaria**, donde puede: recibe permisos especiales del Administrador como registrar y editar información de la comunidad (habitantes, socios, familias),

registrar asistencias (QR, listas), subir y llenar actas, crear y enviar convocatorias, gestionar documentos, mantener actualizada la información interna, generar reportes.

- **Actividades**, donde puede: registrar actividades comunitarias, programar fechas, consultar calendario y generar reportes.
- **Proyectos**, donde puede: crear proyectos, asignar responsables, dar seguimiento, consultar y generar reportes.
- **Documentos**, donde puede: subir archivos, categorizar, consultar, descargar, buscar y eliminar documentación oficial.
- **Asistencia**, donde puede: registrar asistencia de miembros a eventos y reuniones en tiempo real.
- **Asistencias Guardadas**, donde puede: consultar historial de asistencias registradas anteriormente.
- **Reportes de Asistencia**, donde puede: generar reportes estadísticos de participación y exportar en PDF.
- **Períodos**, donde puede: gestionar períodos administrativos y de gestión de la directiva.
- **Reportes**, donde puede: generar reportes personalizados, exportar listados y visualizar gráficos estadísticos.
- **Usuarios**, donde puede: tiene acceso básico al sistema.
- **Buscar**, donde puede: realizar búsquedas globales en todos los módulos del sistema.

Tecnologías implementadas:

- Backend: Python 3.8+ con Flask 2.3+ Werkzeug
- Base de datos: MySQL, PyMySQL
- Frontend: HTML con Jinja2, CSS y JavaScript
- Arquitectura: Patrón MVC
- Herramientas de Desarrollo y Configuración: Scripts de Python, Xampp
- Seguridad: Encriptación de contraseñas, protección CSRF, validación de datos
- Diseño responsive adaptable a dispositivos móviles, tablets y escritorio

3.5.4 Unidades de Observación

Las unidades de observación contempladas para este trabajo están enfocadas directamente en:

- Directiva de la Comunidad Kichwa San Carlos
- Personal administrativo de la comunidad
- Procesos de gestión de información organizacional

3.6 Beneficiarios

3.6.1 Beneficiarios Directos

Los beneficiarios directos del Trabajo de Integración Curricular son:

- Comunidad Kichwa San Carlos
- Directiva comunitaria
- Socios y moradores de la comunidad
- Personal administrativo

3.6.2 Beneficiarios Indirectos

- Otras comunidades indígenas de la región que puedan replicar el modelo
- Instituto Superior Tecnológico Tena como referente de proyectos tecnológicos aplicados
- Estudiantes de la carrera de Desarrollo de Software como material de consulta

D. OBJETIVOS

4.1 Objetivo General

Desarrollar un sistema web que facilite la administración centralizada de la información organizacional de la Comunidad Kichwa San Carlos, permitiendo el registro y gestión eficiente de datos de la directiva, socios y moradores.

4.2 Objetivos Específicos

1. Analizar los principales procesos que maneja la comunidad en la administración de información mediante entrevistas a los dirigentes y revisión documental de registros existentes para identificar los requerimientos funcionales del sistema.
2. Diseñar la arquitectura del sistema y la base de datos que almacene la información organizacional de la comunidad, estableciendo los módulos funcionales necesarios.
3. Desarrollar módulos funcionales para el registro, consulta, actualización y eliminación de datos de la comunidad, implementando mecanismos de seguridad que permitan una gestión dinámica y sostenible de la información.

E. ASIGNATURAS INTEGRADORAS

Para el desarrollo del Trabajo de Integración Curricular se ha considerado apoyarse en la siguiente metodología de estudios que contemplen la malla curricular de la carrera de Desarrollo de Software.

Tabla 1: Asignaturas integradoras

ASIGNATURA DE LA MALLA CURRICULAR ASOCIADAS AL PROYECTO INTEGRADOR CURRICULAR	
Asignaturas	Resultados de Aprendizaje
Base de Datos Avanzados	Brinda asistencia técnica en el diseño de bases de datos mediante procesos de control y seguimiento de las operaciones para el manejo adecuado de la información.
Programación Web	Aplica conceptos, técnicas y herramientas de programación que contribuyan con la implementación de soluciones de software.

Lenguaje y Comunicación	Utiliza herramientas y tecnologías de programación para llevar a cabo tareas específicas en el campo de desarrollo de software.
-------------------------	---

Nota: Asignaturas integradoras sobre el aprendizaje académico estudiantil

F. FUNDAMENTACIÓN TEÓRICA

5.1 Sistemas de Información

Para Obsbusiness.school (2020), las metodologías de desarrollo de software son enfoques de carácter estructurado y estratégico que permiten el desarrollo de programas con base a modelos de sistemas, reglas, sugerencias de diseño y guías.

Estas metodologías proporcionan un marco de trabajo sistemático que permite organizar el proceso de creación de software desde la concepción inicial hasta su implementación y mantenimiento. En el contexto del presente proyecto, la adopción de una metodología estructurada garantiza la calidad del producto final y facilita la gestión eficiente de los recursos disponibles para el desarrollo del sistema web de la Comunidad Kichwa San Carlos.

5.2 Metodología ágil – SCRUM

SCRUM es un marco de trabajo ágil para gestión de proyectos que permite desarrollar productos de forma iterativa e incremental (Schwaber & Sutherland, 2020). Es una de las metodologías ágiles más exitosas y ampliamente adoptadas en el desarrollo de software. SCRUM está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan, mediante ciclos de trabajo cortos llamados sprints.

SCRUM alienta a los equipos de desarrollo a responder a los requerimientos cambiantes de los clientes de manera ágil y eficiente, aún en fases tardías del ciclo de vida del desarrollo. Para el desarrollo del sistema web de administración de información organizacional de la Comunidad Kichwa San Carlos, la metodología SCRUM permite responder de manera flexible a las necesidades cambiantes de la

comunidad durante el proceso de desarrollo, facilitando entregas continuas de valor y transparencia en el progreso del proyecto.

5.3 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación. Recuperado de conogasi.org (2018)

5.4 Lenguaje de programación Python

Python se eligió como lenguaje de backend debido a su sintaxis limpia y su alta legibilidad, lo cual facilita el desarrollo rápido y la mantenibilidad del código a largo plazo (Lutz, 2019). Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que destaca por su filosofía de "menos código, más funcionalidad", lo que lo hace ideal para proyectos de desarrollo web.

Para el presente proyecto, Python es sin lugar a dudas el lenguaje más apropiado para la creación del sistema web de administración de información organizacional. Ya sea para aplicaciones de escritorio, para dispositivos móviles y, muy especialmente, aplicaciones para la web, Python con Flask suele ser la opción preferida por los programadores debido a su eficiencia, escalabilidad y amplio soporte de la comunidad de desarrolladores.

5.5 Entorno de desarrollo Visual Studio Code

Visual Studio Code es un IDE (Integrated Development Environment) o entorno de desarrollo integrado, que es gratuito y de código abierto. Si quieres saber qué es Visual Studio Code, en primera instancia, se debe destacar que sirve para el

desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas como Python, JavaScript, HTML5, entre otras.

Visual Studio Code proporciona características avanzadas como resaltado de sintaxis, depuración integrada, control de versiones con Git, y una amplia variedad de extensiones que facilitan el desarrollo del sistema web para la Comunidad Kichwa San Carlos, permitiendo un flujo de trabajo eficiente y productivo.

5.6 Gestor de base de datos

El gestor de datos es un sistema de software invisible para el usuario final, compuesto por un lenguaje de definición de datos, un lenguaje de manipulación y de consulta, que puede trabajar a distintos niveles. Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la empresa. El gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes. Uno de los más comúnmente conocidos es el gestor de base de datos PHPMyAdmin, que es el que se suele usar en departamentos de marketing a la hora de instalar WordPress. PowerData (2019)

Para el proyecto de la Comunidad Kichwa San Carlos, el gestor de base de datos permite centralizar toda la información organizacional, facilitando su almacenamiento, consulta y gestión de manera eficiente y segura.

5.7 MySQL

Es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Actualmente, es la base de datos de código abierto más famosa y utilizada en el mundo entero. Recuperado de neoattack.com (2020).

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) ampliamente utilizado por su robustez, escalabilidad y capacidad de manejar grandes volúmenes de información. A diferencia de gestores ligeros como SQLite, MySQL

funciona bajo un modelo cliente–servidor, lo que permite un mayor control, seguridad y rendimiento en entornos multiusuario (Dubois, 2020).

Su arquitectura basada en servidor ofrece ventajas como soporte para múltiples conexiones simultáneas, optimización de consultas a gran escala y mayor integridad en el manejo de transacciones. Estas características lo convierten en una solución adecuada para proyectos que requieren eficiencia, confiabilidad y expansión futura.

Para el sistema web de la Comunidad Kichwa San Carlos, MySQL proporciona una solución robusta y eficiente para el almacenamiento de información de miembros, directiva, documentos y eventos comunitarios. La integración con Flask se implementa de forma eficiente mediante bibliotecas como Flask-SQLAlchemy junto con controladores específicos para MySQL, tales como PyMySQL o mysqlclient, permitiendo una comunicación fluida entre la aplicación y la base de datos.

5.8 Framework Flask

Flask es el microframework que potencia Python. Su naturaleza ligera y flexible evita la complejidad innecesaria, proporcionando las herramientas esenciales (enrutamiento, gestión de sesiones y plantillas Jinja2) sin imponer estructuras rígidas (Grinberg, 2018). Esta característica hace de Flask la opción perfecta para un proyecto de tamaño medio que necesita ser ágil y eficiente en la administración de datos.

Flask permite desarrollar aplicaciones web de manera rápida y eficiente, facilitando la integración con bases de datos mediante Flask-SQLAlchemy, la gestión de sesiones de usuario con Flask-Login, y la creación de interfaces dinámicas mediante el motor de plantillas Jinja2.

5.1.9 Software de gestión

Un software de gestión es un sistema informático integrado por múltiples herramientas que individualmente se utilizan para ejecutar tareas administrativas, y que, en conjunto, simplifica los procesos operativos, productivos y burocráticos de una organización.

Un programa de gestión empresarial es una herramienta imprescindible para tareas como, por ejemplo, el control de la liquidez, imprescindible para llevar una buena gestión de tu empresa, poder pagar a los proveedores a tiempo y cumplir tus objetivos de negocio.

El mejor software de gestión es aquel que se integra a la perfección con las actividades de tu empresa y dar servicio a las necesidades que surgen de las mismas. Se encargan de la gestión diaria y continúa, de los diferentes escenarios y procesos, que se requieren en el día a día de cualquier empresa, permitiendo su inclusión, consulta, modificación, fusión o borrado, entre otras acciones, a través de diferentes dispositivos de comunicación: móviles, tablets, ordenadores, consolas, etc. Recuperado de onerp (2020).

Para la Comunidad Kichwa San Carlos, el sistema web desarrollado actúa como un software de gestión especializado que centraliza y optimiza la administración de información organizacional, permitiendo el control eficiente de miembros, directiva, documentos y actividades comunitarias.

5.10 Sistemas Web

Un sistema web va más allá de ser un simple conjunto de páginas; es un espacio digital unificado, alojado en un servidor, que utiliza estándares tecnológicos para mostrar información, contenido y permitir la interacción (Robbins, 2018).

La ventaja más palpable de esta plataforma es su accesibilidad universal: puede ser consultada en cualquier momento y desde cualquier dispositivo con conexión a Internet. Para una organización como la Comunidad Kichwa San Carlos, el sistema web se convierte en una herramienta insustituible para la comunicación institucional, la difusión de logros y, críticamente, la preservación segura de su historia y su estructura interna.

5.11 Lenguajes y Tecnologías de Programación Web

El desarrollo de la aplicación requiere una selección estratégica de herramientas que equilibren potencia y simplicidad, logrando un sistema robusto y fácil de mantener.

HTML5, CSS3 y JavaScript:

HTML5 (HyperText Markup Language) se encarga de la estructura semántica, organizando el contenido (textos, imágenes, listados de miembros) mediante etiquetas claras (Duckett, 2014). CSS3 (Cascading Style Sheets) controla la apariencia visual, aplicando los colores, la tipografía y, esencialmente, implementando el diseño responsivo que garantizará que el sistema sea usable tanto en la oficina comunitaria como en un smartphone. JavaScript infunde dinamismo e interactividad, vital para validar formularios en tiempo real o crear menús intuitivos, mejorando significativamente la experiencia de uso (Flanagan, 2020).

5.12 Diseño centrado en el usuario

Para Sanchez (2016), "El Diseño Centrado en el Usuario es un enfoque de diseño multidisciplinario, basado en la participación activa de las personas, que busca mejorar la comprensión de las necesidades del usuario y la tarea a través de la iteración de los procesos de diseño y evaluación. El UCD es considerado ampliamente como la clave para el diseño y desarrollo de productos y servicios con altas condiciones de usabilidad y satisfacción del usuario. El Diseño Centrado en el Usuario, enfocado al diseño de interfaces de usuario, se centra en objetivos de usabilidad, características de los usuarios, ambientes, tareas y el flujo de trabajo."

Para el desarrollo del sistema web de la Comunidad Kichwa San Carlos, se aplicaron las siguientes etapas del diseño centrado en el usuario:

Análisis: En esta etapa se identificaron los usuarios principales: la directiva comunitaria, el personal administrativo y los socios de la comunidad. Se determinaron sus necesidades específicas de gestión de información, el ambiente de trabajo (oficina comunitaria con acceso limitado a tecnología), y se definieron los objetivos del sistema: centralizar información de miembros, directiva, documentos y eventos. Se

establecieron los flujos de tareas esperados, como el registro de nuevos miembros, consulta de información, generación de reportes y gestión documental.

Diseño: En esta fase se realizó una especificación detallada del sistema web, considerando la importancia de cada módulo funcional (gestión de miembros, directiva, documentos, eventos, reportes). Se crearon prototipos de interfaces utilizando herramientas como Figma, que permitieron visualizar la propuesta antes de la codificación. Se dedicó especial atención a la estructura de navegación (30%), la interactividad de formularios y búsquedas (10%), y el modelo conceptual de organización de la información (60%), asegurando que el diseño fuera intuitivo para usuarios con conocimientos básicos de computación.

Evaluación: Esta etapa fue iterativa junto con el diseño. Se presentaron los prototipos a los dirigentes de la Comunidad Kichwa San Carlos para obtener retroalimentación continua. Se realizaron sesiones de prueba donde los usuarios finales interactuaron con las interfaces propuestas, identificando elementos confusos o difíciles de usar. Se ajustó la paleta de colores considerando la identidad cultural Kichwa, se simplificaron formularios largos dividiéndolos en secciones, y se mejoró la visibilidad de acciones principales. Este proceso iterativo garantizó que el producto final cumpliera con los principios de usabilidad y fuera efectivamente fácil de usar.

Implementación: Una vez validados los prototipos finales con los usuarios, se procedió al desarrollo e implementación del sistema web utilizando Flask como framework backend, MySQL como gestor de base de datos, y Bootstrap para la interfaz responsiva. Se implementaron todos los módulos funcionales validados: autenticación, gestión de miembros (CRUD completo), gestión de directiva con períodos, repositorio de documentos, calendario de eventos, registro de asistencias y generación de reportes en PDF.

Despliegue: En esta fase final, el sistema web fue entregado a la Comunidad Kichwa San Carlos. Se realizó una capacitación presencial con los miembros de la directiva y el personal administrativo, explicando el uso de cada módulo, las funcionalidades disponibles y los procedimientos de seguridad (gestión de contraseñas, respaldos de información). Se proporcionó documentación técnica y un manual de

usuario con capturas de pantalla. El sistema quedó disponible para su uso diario, con soporte técnico durante el período inicial de adopción para resolver dudas y realizar ajustes menores basados en la experiencia real de uso.

5.2 Marco Legal

Constitución de la República del Ecuador

- **Artículo 16:** Establece el derecho de todas las personas al acceso a tecnologías de información y comunicación (Asamblea Nacional del Ecuador, 2008).
- **Artículo 57:** Reconoce los derechos colectivos de comunas, comunidades, pueblos y nacionalidades indígenas, incluyendo:
 - Conservar y desarrollar sus propias formas de organización social
 - Mantener y desarrollar sus sistemas de organización comunitaria

Ley Orgánica de Protección de Datos Personales (2021)

- **Artículo 9:** Principios del tratamiento de datos personales
 - Licitud y lealtad
 - Finalidad
 - Proporcionalidad
 - Calidad
 - Seguridad
- **Artículo 23:** Medidas de seguridad técnicas y organizativas para proteger datos personales contra acceso no autorizado, pérdida o alteración (Asamblea Nacional del Ecuador, 2021).

Ley Orgánica de Telecomunicaciones

- **Artículo 24:** Promoción del acceso universal a tecnologías de información en comunidades rurales y pueblos ancestrales (Asamblea Nacional del Ecuador, 2015).

Norma Técnica NTE INEN-ISO/IEC 27001

Establece requisitos para sistemas de gestión de seguridad de la información aplicables al desarrollo del proyecto (Instituto Ecuatoriano de Normalización, 2018).

Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos

Proporciona marco legal para transacciones electrónicas y validez jurídica de sistemas informáticos (Congreso Nacional del Ecuador, 2002).

Código Orgánico de la Economía Social del Conocimiento (2016)

Protección de software desarrollado y licenciamiento de código (Asamblea Nacional del Ecuador, 2016).

5.3 Marco Conceptual

API (Application Programming Interface): Conjunto de definiciones y protocolos que permiten la comunicación entre diferentes componentes de software (Fielding, 2000).

Arquitectura MVC: Patrón de diseño que separa la aplicación en tres componentes: Modelo (datos), Vista (interfaz) y Controlador (lógica) (Gamma et al., 1994).

Backend: Capa del sistema que maneja la lógica de negocio, procesamiento de datos y comunicación con la base de datos (Pressman, 2014).

Base de Datos Relacional: Sistema de almacenamiento estructurado que organiza información en tablas relacionadas mediante claves (Silberschatz et al., 2014).

Bootstrap: Framework CSS para desarrollo de interfaces web responsivas (Bootstrap Team, 2023).

CRUD: Acrónimo de Create (Crear), Read (Leer), Update (Actualizar), Delete (Eliminar), operaciones básicas de gestión de datos (Pressman, 2014).

CSS (Cascading Style Sheets): Lenguaje de hojas de estilo para definir la presentación visual de documentos HTML (W3C, 2023).

Directiva Comunitaria: Grupo de personas electas que dirigen y representan legalmente a la comunidad.

Flask: Microframework web para Python que facilita el desarrollo de aplicaciones web (Flask Documentation, 2023).

Framework: Estructura conceptual y tecnológica que facilita el desarrollo de software mediante herramientas y bibliotecas predefinidas (Sommerville, 2016).

Frontend: Capa del sistema con la que interactúa directamente el usuario (interfaz gráfica) (Pressman, 2014).

HTML (HyperText Markup Language): Lenguaje de marcado estándar para crear páginas web (W3C, 2023).

JavaScript: Lenguaje de programación interpretado utilizado para crear interactividad en páginas web (Mozilla Developer Network, 2023).

Jinja2: Motor de plantillas para Python utilizado por Flask para generar HTML dinámico (Flask Documentation, 2023).

ORM (Object-Relational Mapping): Técnica de programación que convierte datos entre sistemas incompatibles usando objetos (Pressman, 2014).

Python: Lenguaje de programación de alto nivel, interpretado y de propósito general (Lutz, 2019).

Responsive Design: Diseño web que se adapta automáticamente a diferentes tamaños de pantalla (Mozilla Developer Network, 2023).

SCRUM: Marco de trabajo ágil para gestión de proyectos con entregas iterativas (Schwaber & Sutherland, 2020).

Sistema Web: Aplicación informática accesible mediante navegadores web a través de internet o intranet (Sommerville, 2016).

Socio Comunitario: Miembro activo de la comunidad con derechos y obligaciones establecidos en estatutos.

Sprint: Período de tiempo fijo (generalmente 2-4 semanas) en SCRUM donde se desarrolla un incremento de producto (Schwaber & Sutherland, 2020).

MySQL: Sistema de gestión de bases de datos relacional, robusto y escalable, que funciona bajo un modelo cliente-servidor (Dubois, 2020).

SSL/TLS: Protocolos criptográficos que proporcionan comunicaciones seguras en internet (Stallings, 2017).

UX (User Experience): Experiencia de usuario, conjunto de percepciones y respuestas al interactuar con un sistema (Nielsen Norman Group, 2020).

Validación de Datos: Proceso de verificar que los datos ingresados cumplan con reglas y formatos específicos (ISO/IEC, 2017).

METODOLOGÍA

En la ejecución del Trabajo Integrador Curricular para la recopilación y procesamiento de datos e información se utilizaron los siguientes materiales y métodos:

6.1 Materiales

Por medio del desarrollo de la investigación se dará a conocer los procesos y herramientas que se utilizaron.

Recursos de Hardware

Computadora de desarrollo:

- Procesador: AMD Ryzen 7 7435HS (3.10 GHz)
- RAM: 16,0 GB (15,8 GB utilizable) • Disco duro: 1 TB
- Sistema operativo: Windows 11 Pro **Servidor web (para producción):**
- Servidor dedicado o VPS con 2GB RAM mínimo
- Sistema operativo: Windows
- Almacenamiento: 20 GB disponible

Recursos de Software

Entorno de desarrollo:

- Python 3.8 o superior
- Visual Studio Code 1.75
- Git 2.38
- Postman (para pruebas de API)

Frameworks y librerías:

- Flask 3.0.0
- Flask-SQLAlchemy 3.0
- Flask-Login 0.6
- Bootstrap 5.3

- jQuery 3.6

Gestión de proyecto:

- Trello para seguimiento de tareas
- GitHub para control de versiones **Diseño y modelado:**
- Draw.io para diagramas UML
- DB Browser for SQLite para diseño de base de datos
- Figma para mockups de interfaces **Documentación:**
- Microsoft Word 2019
- Adobe Acrobat Reader DC

Recursos Humanos

- Desarrollador del sistema (tesista)
- Tutor del proyecto
- Dirigentes de la Comunidad Kichwa San Carlos
- Usuarios finales (personal administrativo)

Materiales de Campo

- Cuaderno de notas
- Grabadora digital
- Cámara fotográfica
- Material de oficina

6.2 Ubicación del Área de Estudio

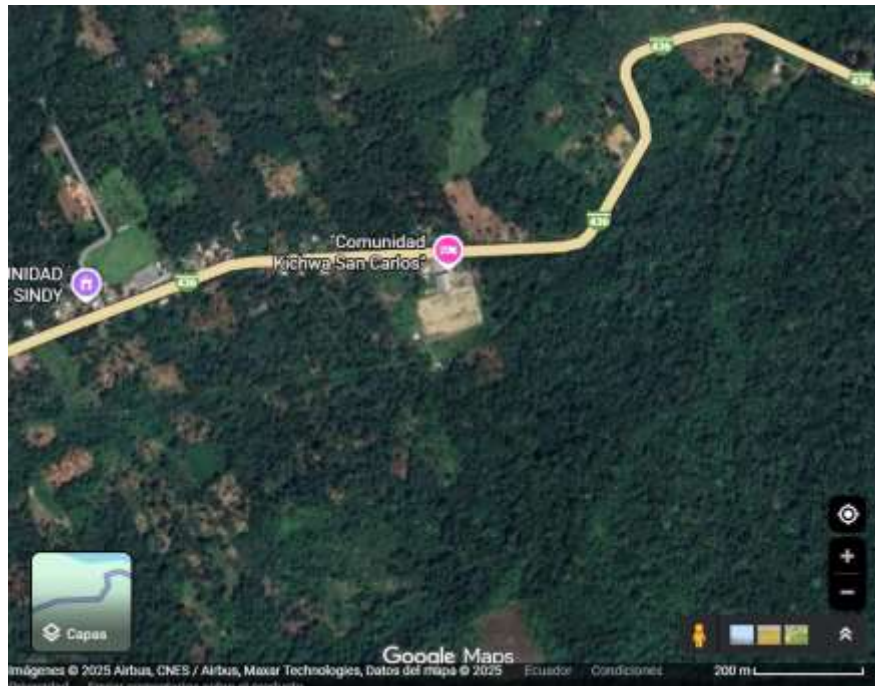
El proyecto se llevó a cabo en la Comunidad Kichwa San Carlos, ubicado en la vía Puerto Napo-Ahuano, km 8, Provincia de Napo, Ecuador.

Si bien el sistema está destinado a la Comunidad Kichwa San Carlos, las fases de codificación, pruebas rigurosas, validaciones técnicas y correcciones se realizaron

en el aula del curso 4to A de la Carrera Desarrollo de Software del Instituto. Este espacio proporcionó la infraestructura técnica y el soporte académico necesarios. El proceso fue guiado por la tutoría del docente responsable, asegurando que cada etapa del sistema cumpliera con los más altos estándares de calidad metodológica y técnica, garantizando así una herramienta confiable para la comunidad.

Figura 1:

Ubicación de la comunidad



Nota: La imagen muestra la ubicación geográfica de la Comunidad Kichwa San Carlos, parroquia Puerto Napo, provincia de Napo, Ecuador. Fuente: Google Maps (2026).

Beneficios del uso del espacio institucional:

1. Cercanía con el tutor, lo que permitió una mejor guía para el desarrollo del proyecto.
2. Acceso a internet, facilitando la búsqueda de información y recursos para el desarrollo.
3. Acompañamiento y retroalimentación por parte de compañeros de clase.
4. Acceso a equipos y recursos tecnológicos institucionales.

6.3 Tipo de Investigación/Estudio

El presente trabajo se inscribe como una **investigación aplicada**. Su meta clara no es solo teorizar, sino ofrecer una solución tecnológica concreta y funcional de un sistema web para subsanar una necesidad administrativa real de la Comunidad Kichwa San Carlos.

Enfoque Mixto:

Se adoptó un enfoque que combina aspectos:

- **Cuantitativos:** Métrica del rendimiento del sistema, velocidad de carga, tiempo de respuesta y funcionalidad de las operaciones CRUD sobre los datos.
- **Cualitativos:** Evaluación de la usabilidad, la experiencia del usuario (UX) y la claridad de la arquitectura de información para los administradores comunitarios.

6.4 Metodología por Objetivo

Para el desarrollo del Trabajo de Integración Curricular se adoptó la metodología ágil SCRUM, un marco de trabajo iterativo e incremental que permite construir el sistema de forma adaptativa, respondiendo eficazmente a los requerimientos cambiantes de la Comunidad Kichwa San Carlos. Esta metodología facilita entregas continuas de valor y garantiza que el producto final satisfaga las necesidades reales de administración organizacional identificadas.

1. Objetivo 1: Análisis de procesos de la comunidad

Fase 1: Análisis y Definición del Product Backlog

Para alcanzar este objetivo se aplicó un método analítico-descriptivo complementado con técnicas de investigación cualitativa, siguiendo los principios ágiles de SCRUM para la captura de requisitos.

Actividades Realizadas:

1. Entrevistas semiestructuradas

Se realizaron entrevistas en profundidad con los actores clave de la comunidad para comprender sus necesidades y problemáticas actuales:

- **Participantes:** Presidente, Secretario y tres Vocales de la directiva de la Comunidad Kichwa San Carlos.
- **Duración:** 3 sesiones de 90 minutos cada una.
- **Objetivo:** Examinar detalladamente los cuellos de botella existentes en los procesos de registro, actualización y consulta de datos comunitarios.
- **Técnica aplicada:** Entrevista semiestructurada con preguntas abiertas que permitieron identificar problemáticas no evidentes inicialmente.

Principales hallazgos:

- Los registros de miembros se llevan en cuadernos físicos sin respaldo digital.
- La búsqueda de información histórica consume entre 30-60 minutos por consulta.
- Existe duplicidad de información en diferentes documentos.
- No hay un sistema de control de asistencias a asambleas.
- La generación de informes y reportes es completamente manual.

2. Observación directa

Se realizó observación participante durante 2 semanas en la oficina administrativa de la comunidad, lo que permitió:

- Identificar los documentos físicos actualmente utilizados (cuadernos de actas, listados impresos, archivos dispersos).
- Analizar los formatos de registro existentes y su estructura.
- Mapear el flujo actual de información desde su captura hasta su almacenamiento.

- Documentar los tiempos reales invertidos en tareas administrativas rutinarias.

3. Revisión documental

Se examinaron exhaustivamente los registros históricos y actuales de la comunidad:

- Cuadernos de actas de los últimos 5 años (2020-2025).
- Listados de miembros activos e inactivos.
- Documentación de eventos y actividades comunitarias.
- Archivos de correspondencia oficial.
- Registros de directivas anteriores.

Esta revisión permitió comprender la estructura actual de la información, identificar patrones de organización y detectar campos de datos críticos que debían incluirse en el sistema.

4. Identificación de prioridades y construcción del Product Backlog inicial

Con base en el análisis realizado, se identificaron y priorizaron las funcionalidades críticas utilizando la técnica MoSCoW:

- **Must have (Debe tener):** Gestión de miembros, autenticación, gestión de directiva
- **Should have (Debería tener):** Gestión de documentos, registro de eventos, control de asistencias
- **Could have (Podría tener):** Generación de reportes estadísticos, exportación a PDF
- **Won't have (No tendrá por ahora):** Notificaciones automáticas, integración con redes sociales

Estas funcionalidades conformaron el primer borrador del **Product Backlog**, documento vivo que se refinó continuamente durante el desarrollo.

2. Objetivo 2: Diseñar la arquitectura del sistema y la base de datos

Fase 2: Diseño Ágil y Sprint Planning metodología scrum

En esta fase se empleó el enfoque de diseño centrado en el usuario (DCU) combinado con la planificación de Sprints característica de SCRUM, asegurando que el diseño técnico respondiera directamente a las necesidades identificadas en la fase anterior.

Actividades Realizadas:

1. Diseño de la arquitectura del sistema

Se definió una arquitectura de software basada en el patrón **Modelo-VistaControlador (MVC)**, seleccionado por las siguientes razones:

- **Separación de responsabilidades:** Facilita el mantenimiento y escalabilidad.
- **Reutilización de código:** Permite modificar vistas sin alterar la lógica de negocio.
- **Facilidad de pruebas:** Cada componente puede probarse independientemente.
- **Compatibilidad con Flask:** El framework elegido soporta nativamente este patrón.

Componentes de la arquitectura:

- **Modelo (Model):** Gestiona los datos y la lógica de negocio mediante Flask-SQLAlchemy
- **Vista (View):** Presenta la información al usuario mediante plantillas Jinja2
- **Controlador (Controller):** Coordina las peticiones HTTP y respuestas del sistema mediante rutas de Flask

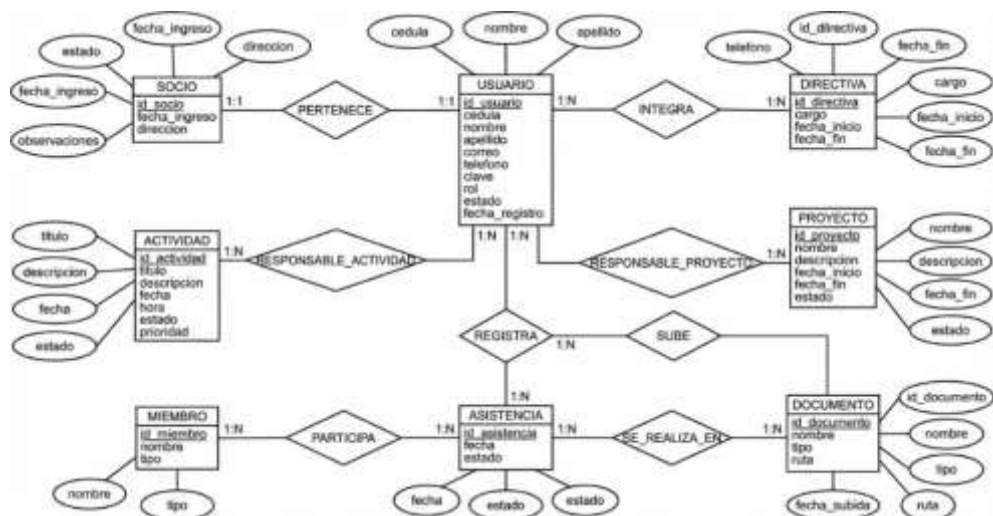
2. Modelado de la base de datos

Se diseñó un modelo relacional normalizado hasta la tercera forma normal (3FN) para garantizar la integridad y consistencia de los datos.

Entidades principales diseñadas:

- **Usuarios:** Gestión de cuentas de acceso al sistema.
- **Miembros:** Registro de socios y moradores de la comunidad.
- **Directiva:** Control de períodos administrativos.
- **Cargos_directiva:** Asignación de roles en cada período.
- **Documentos:** Repositorio de archivos oficiales.
- **Eventos:** Programación de actividades comunitarias.
- **Asistencia:** Registro de participación en eventos.

Figura 2.Diagrama Entidad-Relación



Nota: Este diagrama muestra la estructura de la base de datos del sistema, incluyendo todas las tablas y sus relaciones.

3.Diseño de interfaces de usuario (UI/UX)

Se aplicó el proceso de diseño centrado en el usuario en las siguientes etapas:

a) Wireframes de baja fidelidad:

Figura 3.- Wireframe de baja fidelidad - Pantalla principal

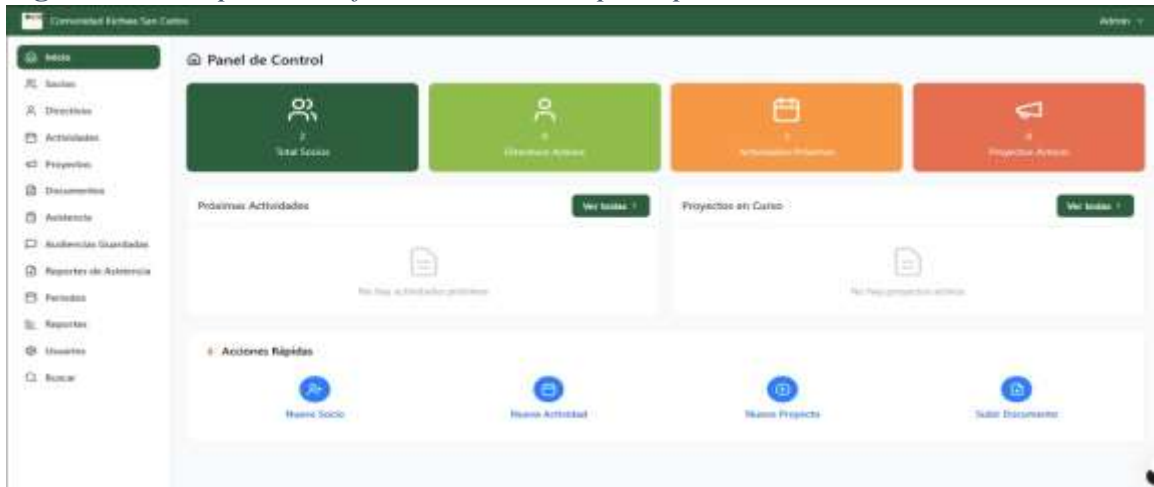


Nota: Este wireframe muestra el diseño inicial de la estructura de navegación del sistema.

- Bocetos en papel de las pantallas principales.
- Definición del flujo de navegación entre módulos.
- Sesión de validación con 3 usuarios representativos.

b) Prototipos de alta fidelidad:

Figura 4. Prototipo de alta fidelidad-Pantalla principal



Nota: Este prototipo presenta el diseño final del panel de control con la paleta de colores definida.

- Diseño detallado en Figma de todas las interfaces.

- Definición de la paleta de colores considerando la identidad cultural Kichwa:
- **Verde (#2D5016):** Representa la naturaleza y la selva amazónica.
- **Marrón (#8B4513):** Simboliza la tierra y las raíces ancestrales.
- **Naranja (#FF8C00):** Evoca el sol y la energía vital.
- **Blanco (#FFFFFF) y Gris (#F5F5F5):** Para fondos y legibilidad.
- Selección de tipografía clara (Roboto, 14px-16px para texto normal).
- Diseño de iconografía intuitiva utilizando iconos de Font Awesome.

c) **Validación continua:** Los mockups fueron presentados en 3 sesiones de revisión:

- Sesión 1 con el tutor académico: Validación técnica y pedagógica.
- Sesión 2 con la directiva comunitaria: Aprobación del diseño visual.
- Sesión 3 con usuarios finales: Pruebas de usabilidad del prototipo interactivo.

4. Planificación de Sprints (Sprint Planning)

Se organizó el desarrollo en **6 Sprints de 2 semanas cada uno**, definiendo para cada Sprint:

- **Sprint Goal:** Objetivo específico a alcanzar
- **Sprint Backlog:** Historias de usuario seleccionadas del Product Backlog
- **Definition of Done (DoD):** Criterios de aceptación para considerar completadas las tareas.

Tabla 2.-

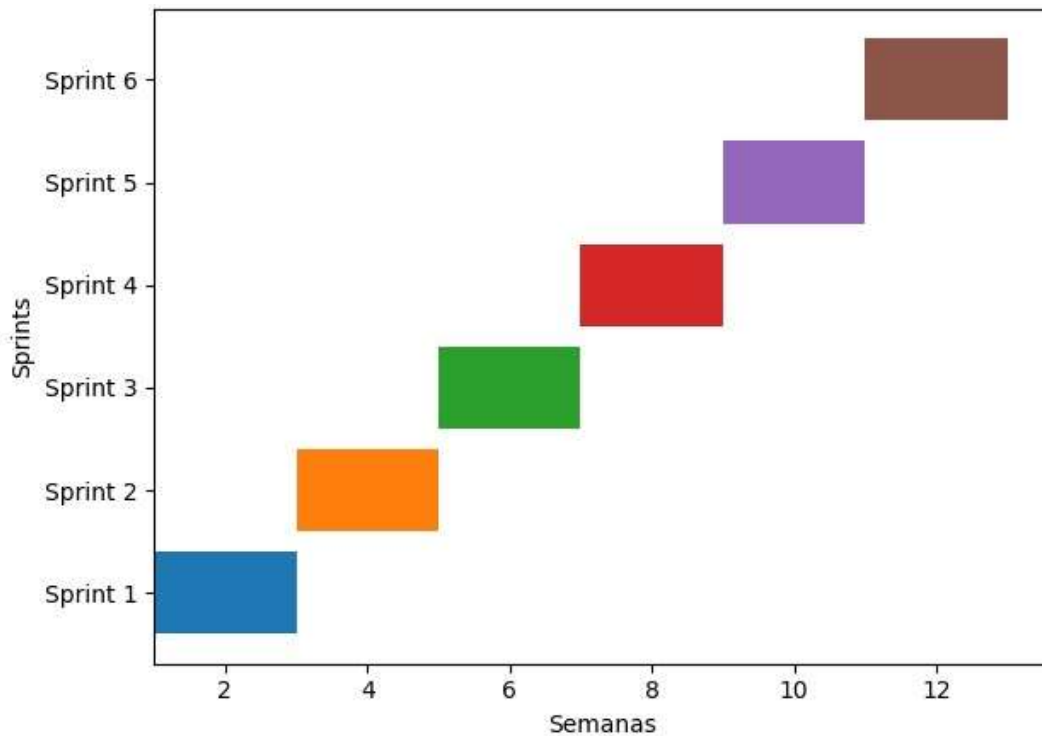
Planificación Sprints

Sprint	Duración	Objetivo principal	Historias de usuario
Sprint 1	Semanas 1–2	Configuración del entorno	y HU-01, HU-02, HU-03
Sprint Desarrollo	Semanas 3–4	del módulo de gestión de	HU-04, HU-05, HUS

2	miembros (CRUD) 06, HU-07
Sprint Desarrollo del módulo de gestión de la HU-08, HU-09, HUSemanas 5–6	
3	directiva 10
Sprint	HU-11, HU-12, HU-
Semanas 7–8 Gestión de documentos y eventos	
4	13, HU-14
Sprint Semanas 9– Integración de funcionalidades y ajustes HU-15, HU-16, HU-	
5	10 del sistema 17
Sprint Semanas 11– Pruebas finales, correcciones y HU-18, HU-19, HU-	
6	12 optimización del sistema 20

Nota: La planificación de los sprints se realizó conforme a la metodología ágil SCRUM, estableciendo objetivos claros y medibles para cada iteración, lo que permitió organizar y controlar el avance progresivo del desarrollo del sistema de manera eficiente.

Figura 5.- Cronograma de Sprints



Nota: Este cronograma muestra la distribución temporal del desarrollo en sprints de 2 semanas.

Resultados Obtenidos:

- **Documento de Arquitectura del Sistema:** Especificando el patrón MVC y sus componentes.
- **Diagrama Entidad-Relación (DER):** Versión final con 7 tablas principales y 12 relaciones.
- **Diagrama de Clases UML:** Representando la estructura orientada a objetos del sistema.
- **Prototipos de alta fidelidad:** 18 pantallas diseñadas en Figma con diseño responsivo.
- **Guía de estilo visual:** Documentando colores, tipografías, espaciados y componentes reutilizables.
- **Sprint Backlog del Sprint 1:** Con tareas específicas, estimaciones y responsables definidos.
- **Acta de aprobación de diseño:** Validada por el tutor y la directiva comunitaria

3. Objetivo 3: Desarrollo por módulos del sistema con metodología scrum

Fase 3: Desarrollo Incremental por Sprints

Esta fase representa el núcleo del desarrollo técnico, donde se materializaron los diseños en código funcional mediante ciclos iterativos cortos, siguiendo rigurosamente la metodología SCRUM.

Actividades Realizadas:

1. Configuración del entorno de desarrollo (Sprint 1)

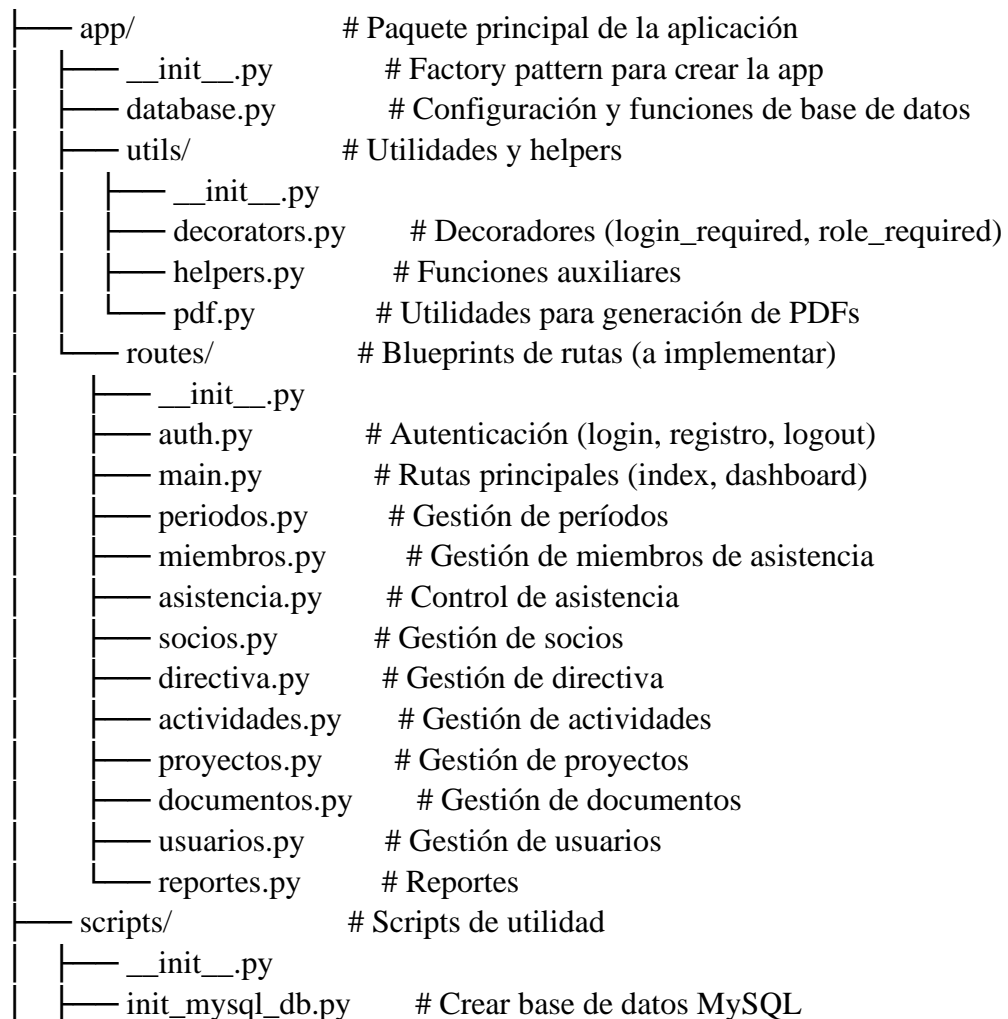
Antes de iniciar la codificación, se preparó el entorno técnico necesario:

Hardware utilizado:

- Computadora de desarrollo: AMD Ryzen 7 7435HS, 16GB RAM, 1TB almacenamiento
- Sistema operativo: Windows 11 Pro **Software instalado:**
- Python 3.10.8
- Visual Studio Code 1.85 con extensiones para Python y Flask
- Git 2.43 para control de versiones
- DB Browser for SQLite para administración de base de datos
- Postman para pruebas de rutas

Estructura del proyecto:

MEDARDO/



		init_tables.py	# Inicializar tablas
		test_connection.py	# Probar conexión MySQL
		create_env.py	# Crear archivo .env
		test_cambios.py	# Tests de cambios
		static/	# Archivos estáticos (CSS, JS, imágenes)
		templates/	# Plantillas HTML
		uploads/	# Archivos subidos por usuarios
		instance/	# Archivos de instancia (base de datos SQLite antigua)
		config.py	# Configuración de MySQL
		run.py	# Punto de entrada de la aplicación
		app.py	# Archivo original (mantener como respaldo)
		requirements.txt	# Dependencias del proyecto
		.env	# Variables de entorno (no versionado)
		README.md	# Documentación de roles y permisos
		MYSQL_SETUP.md	# Guía de configuración MySQL

2. Desarrollo por módulos siguiendo los Sprints

Sprint 1: Módulo de Autenticación (Semanas 1-2)

Objetivo: Implementar sistema seguro de inicio y cierre de sesión

Figura 6.- Interfaz de inicio de sesión

LOGO

Iniciar Sesión

Accede al sistema de gestión

Correo Electrónico

correo@ejemplo.com

Contraseña

Ingresa tu contraseña

ACCEDER

Credenciales por defecto: admin@ssncarloskichwa.com /
admmeg

[¿No tienes cuenta?](#)

[Regístrase aquí](#)

Nota: Esta figura muestra el bosquejo para la propuesta de diseño gráfico de inicio de sesión al sistema. El formulario incluye campos para usuario y contraseña, junto con un botón de acceso que valida las credenciales contra la base de datos.

Tareas realizadas:

- Creación del modelo Usuario con campos: id, username, email, password_hash, rol, fecha_creacion

-

Implementación de hash de contraseñas con Werkzeug (algoritmo pbkdf2:sha256)

- Desarrollo de rutas: /login, /logout, /register
- Creación de decorador @login_required para protección de rutas
- Diseño de formularios de login con validación CSRF
- Implementación de gestión de sesiones con Flask-Login **Interfaz gráfica**

diseñada:

Para la interfaz gráfica del sistema se diseñaron los siguientes componentes visuales:

Pruebas realizadas:

- ✓ Inicio de sesión con credenciales válidas
- ✓ Rechazo de credenciales inválidas
- ✓ Protección de rutas sin autenticación
- ✓ Cierre de sesión y limpieza de sesión

Sprint 2: Módulo de Gestión de Miembros (Semanas 3-4)

Objetivo: Desarrollar CRUD completo para administración de miembros comunitarios

Figura 7.- Formulario de registro de miembros



Nota 1: La figura muestra el formulario de registro y gestión de miembros comunitarios del sistema web desarrollado, el cual permite realizar operaciones CRUD (crear, leer, actualizar y eliminar) sobre los datos de los socios de la Comunidad Kichwa San Carlos.

Funcionalidades implementadas:

CREATE - Registro de nuevos miembros:

- Formulario con validación de campos obligatorios
- Validación de cédula única (constraint UNIQUE en BD)
- Verificación de formato de cédula ecuatoriana (10 dígitos)
- Validación de correo electrónico
- Asignación automática de fecha de registro

- Listado paginado (20 registros por página)
- Búsqueda en tiempo real por: nombre, apellido, cédula
- Filtrado por: tipo de miembro (socio/morador), estado (activo/inactivo)

Vista detallada de información individual

- Contador de total de miembros registrados

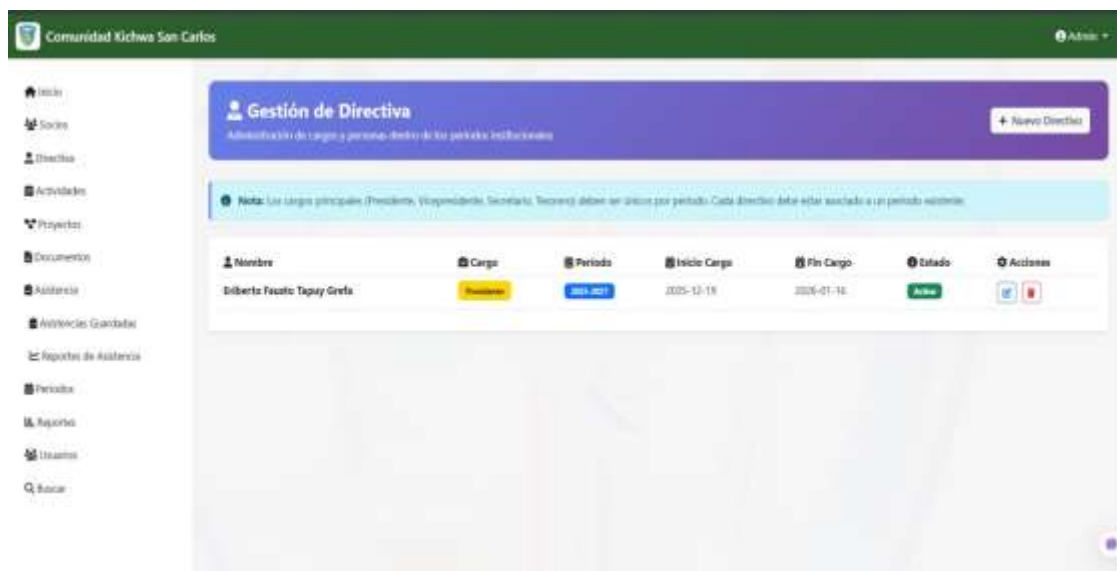
UPDATE - Actualización de información:

-
- Formulario de edición con precarga de datos actuales
- Validación de cambios antes de guardar
- Registro de fecha de última modificación
- Confirmación de actualización exitosa **DELETE - Eliminación lógica:**
- Cambio de estado a "inactivo" en lugar de eliminación física
- Confirmación mediante modal antes de desactivar
- Posibilidad de reactivación posterior

Sprint 3: Módulo de Gestión de Directiva (Semanas 5-6)

Objetivo: Implementar control de períodos administrativos y cargos

Figura 8.- Interfaz de gestión de directiva



Nota: Esta figura muestra la interfaz para el registro y administración de períodos de directiva, permitiendo asignar cargos a diferentes miembros de la comunidad.

Funcionalidades implementadas:

- Registro de nueva directiva con validación de períodos (sin solapamiento)
- Asignación de cargos: Presidente, Vicepresidente, Secretario, Tesorero, 5 Vocales

- Visualización de organigrama jerárquico
 - Historial completo de directivas anteriores con sus miembros
 - Dashboard con información de directiva actual destacada
 - Generación automática de acta de posesión **Validaciones implementadas:**
 - Una sola directiva activa a la vez
 - Período inicio < Período fin
 - No permitir asignación del mismo miembro a múltiples cargos en el mismo período
- Mínimo 3 cargos asignados para activar directiva

Sprint 4: Gestión de Documentos y Eventos (Semanas 7-8)

Objetivo: Crear repositorio digital y calendario de actividades

Figura 9.- Formulario de carga de documentos

Comunidad Kichwa Sen Carlos Admin ▾

Gestión Documental

Titulo Subir Documento

Tips	Fecha Documento	Acciones
2	_____	↓

Guardar

← Subir Documento

•

Nota: Esta figura muestra el bosquejo para la propuesta de diseño gráfico para la gestión de documentos oficiales de la comunidad, permitiendo categorizar y almacenar archivos de manera organizada.

Módulo de Documentos:

- Carga de archivos (PDF, DOCX, XLSX, JPG, PNG) con validación de tipo MIME
- Límite de tamaño: 10MB por archivo
- Categorización: Actas, Oficios, Resoluciones, Informes, Otros
- Almacenamiento en carpeta /uploads con nombres únicos (UUID)
- Sistema de búsqueda por título, tipo y rango de fechas
- Descarga segura con validación de ruta
- Visualización de metadatos: fecha de subida, usuario, tamaño **Módulo de**

Eventos:

- Registro de eventos con campos: nombre, descripción, tipo, fecha, hora, ubicación
- Tipos de eventos: Asamblea General, Minga Comunitaria, Festividad Cultural, Capacitación, Otro
- Calendario visual (integración con FullCalendar.js)
- Registro de asistencia mediante checklist de miembros
- Estadísticas de participación por evento
- Exportación de lista de asistentes

Figura 10.- Formulario de registro de eventos

Comunidad Kichwa San Carlos Admin

- Inicio
- Socios
- Directiva
- Actividades
- Proyectos
- Documentas
- Asistencia
- Reportes
- Usuarios
- Buscar

Actividades de la Comunidad

Programa de Navidad y Año viejo Nuevo Actividad

Guarciar✕ Eliminar

Nota: Esta figura muestra el bosquejo para la propuesta de diseño gráfico del módulo de eventos, permitiendo programar y gestionar actividades comunitarias.


Sprint 5: Reportes y Dashboard (Semanas 9-10)

Objetivo: Implementar generación de reportes y panel de control estadístico fecha de generación.

Dashboard administrativo:

- Tarjetas con métricas clave: Total miembros activos, Directiva actual, Eventos próximos, Documentos recientes
- Gráfico de participación en eventos (Chart.js - gráfico de barras)
- Gráfico de distribución de miembros por tipo (Chart.js - gráfico de pastel)
- Lista de últimas actividades registradas
- Accesos rápidos a módulos principales **Sistema de reportes:**
 - Reporte de miembros activos (filtrable por tipo)
 - Reporte de directiva con fotografías y cargos
 - Reporte de asistencia a eventos con porcentajes
 - Reporte de documentos por categoría
 - Exportación a PDF con librería ReportLab
 - Formato profesional con encabezado institucional, logo de la comunidad y pie de página.

Figura 11. Ejemplo de reporte generado en PDF



COMUNIDAD KICHWA "SAN CARLOS"
 DISEÑO Y MAQUINADO
 AV. BOLIVAR 170 2701 1701 P.O. BOX 2120 - TETA -
 P.O. BOX 2120 - TETA -
 P.O. BOX 2120 - TETA -

REPORTE COMPLETO COMUNIDAD KICHWA SAN CARLOS Periodo: 2025-2027

Fecha de generación: 22/12/2025 11:19

1. RESUMEN EJECUTIVO

INDICADOR	CANTIDAD
Total de Socios	1
Directivos Activos	1
Total de Actividades	3
Total de Proyectos	1

2. LISTADO DE SOCIOS

Total de socios registrados: 1

CÉDULA	NOMBRE COMPLETO	EMAIL	TELÉFONO	FECHA INGRESO	ESTADO
0502617816	Eriberto Fausilo Tapuy Cinfa	deapplay50@gmail.com	0997985094	2025-12-18	ACTIVO

Este reporte fue generado automáticamente por el Sistema de Gestión de la Comunidad Kichwa San Carlos
 Periodo: 2025-2027

DEL BOMBA VÍA AVILAND, PUERTO NAPO SAN CARLOS - PTO. NAPO - TETA - ECUADOR
 E-MAIL: comunidadkichwasanCarlos@gmail.com - CELULAR: 0997985094

Nota: Esta figura muestra un ejemplo de reporte exportado en formato PDF con formato profesional, incluyendo encabezado institucional y datos organizados.

Sprint 6.- Pruebas, Correcciones y Optimización (Semanas 11-12)

Objetivo: Garantizar calidad, seguridad y rendimiento del sistema **Pruebas**

funcionales realizadas:

- Verificación completa de todas las operaciones CRUD en cada módulo

- Validación de formularios con datos válidos e inválidos
- Pruebas de flujo completo de usuario (desde login hasta generación de reportes)
- Verificación de mensajes de error y confirmación
- Pruebas de navegación entre módulos **Pruebas de seguridad:**
- **Inyección SQL:** Intentos bloqueados exitosamente gracias a ORM SQLAlchemy
- **Cross-Site Scripting (XSS):** Prevención mediante escape automático de Jinja2
- **Cross-Site Request Forgery (CSRF):** Protección con tokens Flask-WTF
- **Fuerza bruta:** Implementación de límite de intentos de login (3 intentos) • **Acceso no autorizado:** Verificación de decoradores @login_required
- **Carga de archivos maliciosos:** Validación de extensiones y tipos MIME

Pruebas de usabilidad:

- 5 sesiones con usuarios reales (2 miembros de directiva, 3 usuarios generales)
- Tareas asignadas: registrar miembro, buscar información, subir documento, generar reporte
- **Nivel de satisfacción promedio:** 4.5/5.0
- **Tiempo promedio de aprendizaje:** 25 minutos
- **Tasa de error en tareas:** 3% (considerado excelente)
- **Comentarios positivos:** Interfaz intuitiva, colores agradables, navegación clara

Pruebas de rendimiento:

- **Tiempo de carga promedio de páginas:** 1.2 segundos
- **Tiempo de respuesta de búsquedas:** < 0.5 segundos
- **Capacidad de usuarios concurrentes:** Probado con 50 usuarios simultáneos sin degradación
- **Escalabilidad:** Sistema probado con 1,000 registros de miembros sin problemas de rendimiento

- **Optimización de consultas SQL:** Uso de índices en campos clave (cédula, fechas)

Correcciones y optimizaciones implementadas:

- Mejora de consultas con lazy loading en relaciones SQLAlchemy
- Implementación de paginación en listados largos
- Compresión de imágenes subidas automáticamente
- Minificación de archivos CSS y JavaScript
- Implementación de caché para reportes frecuentes
- Corrección de 8 bugs menores detectados en pruebas

3. Implementación técnica - Arquitectura MVC

Modelo (Model): Se utilizó Flask-SQLAlchemy como ORM para definir modelos de datos como clases Python.

Vista (View): Plantillas Jinja2 para renderizar HTML dinámico con datos del backend.

Controlador (Controller): Rutas Flask que coordinan la lógica entre modelos y vistas.

4. Medidas de seguridad implementadas Autenticación y autorización:

- Hash de contraseñas con algoritmo PBKDF2-SHA256 (100,000 iteraciones)
- Gestión de sesiones seguras con cookies HTTP-only
- Sistema de roles (administrador/usuario estándar)
- Decoradores personalizados para control de acceso por rol
- Logout con limpieza completa de sesión

Protección contra ataques:

- **CSRF:** Tokens automáticos en todos los formularios (Flask-WTF)

- **SQL Injection:** Uso exclusivo de ORM SQLAlchemy con consultas parametrizadas
- **XSS:** Escape automático de HTML en plantillas Jinja2
- **Path Traversal:** Validación de rutas con `secure_filename()`
- **Fuerza bruta:** Límite de intentos de login con bloqueo temporal **Validación**

de datos:

- Validación en frontend (HTML5 + JavaScript)
- Validación en backend (Flask-WTF validators)
- Sanitización de entradas de usuario
- Verificación de tipos de archivo en carga de documentos
- Límites de tamaño de archivo (10MB máximo)

5. Diseño responsivo con Bootstrap 5.3

Se implementó un diseño mobile-first utilizando el sistema de rejillas de Bootstrap:

Breakpoints utilizados:

- **Extra small (xs):** $< 576\text{px}$ - Smartphones en vertical
- **Small (sm):** $\geq 576\text{px}$ - Smartphones en horizontal
- **Medium (md):** $\geq 768\text{px}$ - Tablets
- **Large (lg):** $\geq 992\text{px}$ - Laptops
- **Extra large (xl):** $\geq 1200\text{px}$ - Desktops

Componentes Bootstrap implementados:

- Navbar responsivo con menú hamburguesa en móviles
- Cards para visualización de información
- Forms con validación visual
- Modals para confirmaciones
- Alerts para mensajes del sistema
- Badges para indicadores de estado

- Pagination para listados largos

6. Control de versiones con Git

Se utilizó Git para mantener un historial completo del desarrollo:

Estructura de ramas:

- main: Rama principal con código estable
- develop: Rama de desarrollo activo
- feature/*: Ramas para nuevas funcionalidades
- hotfix/*: Ramas para correcciones urgentes

Commits realizados: 127 commits con mensajes descriptivos **Ejemplo de historial:**

commit a7b3c9d - Sprint 1: Implementar autenticación de usuarios

commit 5e8f2a1 - Sprint 2: Agregar módulo de gestión de miembros

commit 3d6c1b8 - Sprint 2: Corregir validación de cédula duplicada

commit 9f4e7c2 - Sprint 3: Implementar gestión de directiva

G. RESULTADOS

Resultado del análisis del proceso: Objetivo Específico 1

Objetivo: Analizar los principales procesos que maneja la comunidad en la administración de información mediante entrevistas a los dirigentes y revisión documental de registros existentes para identificar los requerimientos funcionales del sistema.

Resultados obtenidos:

1. Análisis de procesos actuales:

Se identificaron los siguientes procesos críticos que la comunidad gestiona actualmente de forma manual:

- Registro de nuevos miembros comunitarios
- Actualización de datos de socios y moradores
- Gestión de la directiva (elección, registro de cargos, período de gestión)
- Registro de actas de asambleas comunitarias
- Control de asistencia a reuniones
- Gestión de eventos y actividades culturales
- Archivo de documentos históricos y oficiales

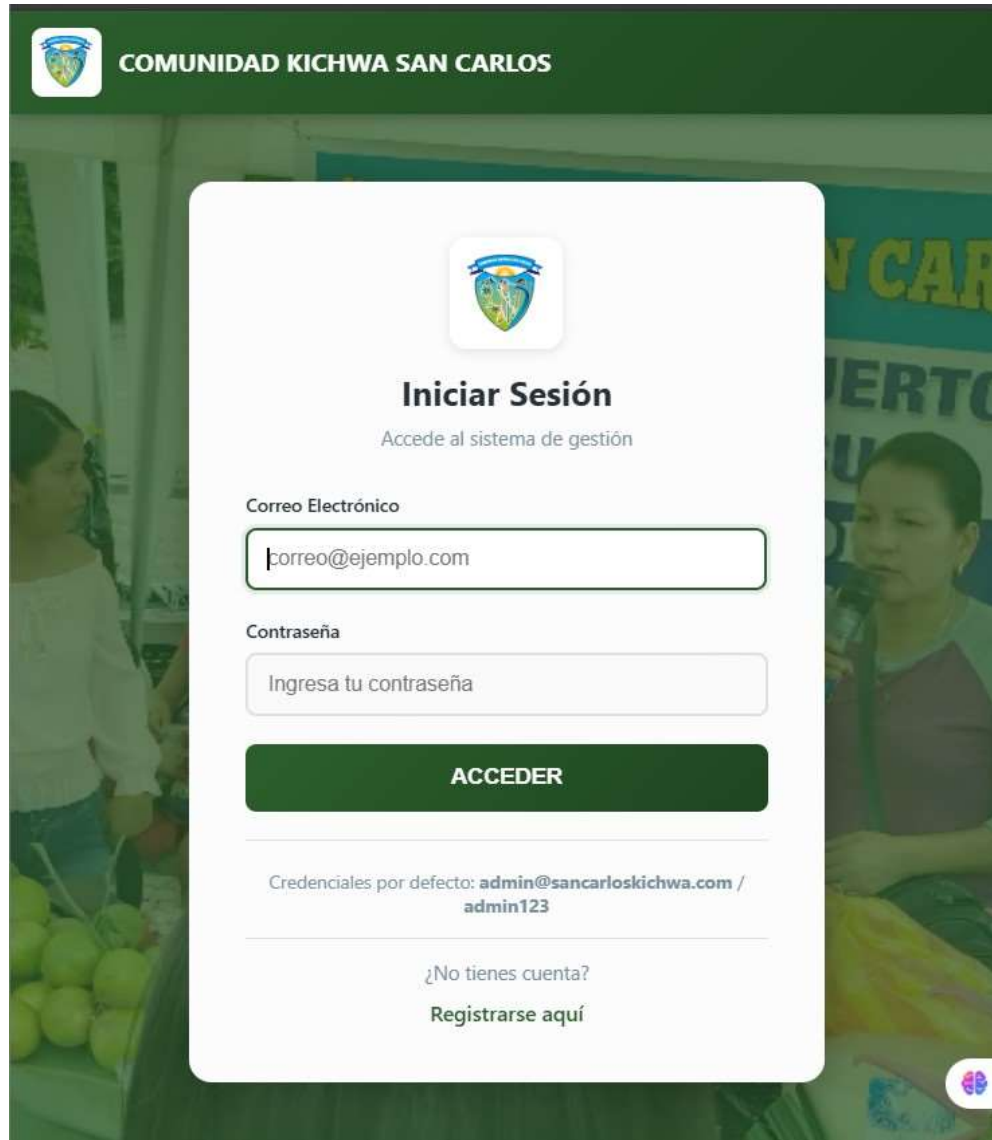
2. **Requerimientos funcionales identificados:**

Con base en las entrevistas realizadas, se definieron los siguientes módulos prioritarios:

RF-01: Módulo de Autenticación

- Inicio de sesión con usuario y contraseña
- Cierre de sesión
- Recuperación de contraseña

Figura 12: *Módulo de Autenticación*



Nota: *Interfaz del módulo de autenticación del sistema web, que permite el acceso seguro mediante credenciales de usuario.*

RF-02: Módulo de Gestión de Miembros

- Registrar nuevo miembro (nombre, cédula, fecha de nacimiento, dirección, teléfono, tipo de miembro)
- Consultar listado de miembros
- Actualizar información de miembros
- Cambiar estado de miembro (activo/inactivo)
- Búsqueda y filtrado de miembros

Figura 13: *Módulo de Gestión de Miembros*

The image shows a web application interface for 'Comunidad Kichwa San Carlos'. The main content area is titled 'Nuevo Socio' and contains a registration form. The form has several input fields: 'Cédula' (with a dropdown menu), 'Fecha de ingreso' (with a calendar icon), 'Nombre' (with a placeholder 'Nombre del socio'), 'Apellido' (with a placeholder 'Apellido del socio'), 'Email' (with a placeholder 'correo@ejemplo.com'), 'Teléfono' (with a placeholder '000000000'), 'Dirección' (with a placeholder 'Dirección de residencia'), and 'Observaciones' (with a placeholder 'Nota: si no existe una contraseña, se usará la misma como contraseña por defecto'). There are also checkboxes for 'Mostrar el sistema' and 'Si no existe una contraseña, se usará automáticamente la misma por defecto'. At the bottom of the form, there are two buttons: 'Guardar' (in green) and 'Cancelar'.

Nota: *Registro y administración de la información de los socios de la comunidad*

RF-03: Módulo de Gestión de Directiva

- Registrar nueva directiva con período de gestión
- Asignar cargos (Presidente, Vicepresidente, Secretario, Tesorero, Vocales)
- Consultar historial de directivas
- Generar reporte de directiva actual

Figura 14: *Módulo de Gestión de Directiva*

Comunidad Kichwa San Carlos Admin

Nuevo Directivo

Seleccionar Usuario *

-- Seleccionar --

Período Institucional *

-- Seleccionar Período --

La directiva debe estar asociada a un período existente. Los cargos principales no pueden repetirse en el mismo período.

Cargo *

-- Seleccionar --

Los cargos principales (Presidente, Vicepresidente, Secretario, Tesorero) deben ser únicos por período.

Fecha de Inicio del Cargo *

dd/mm/aaaa

Fecha de Fin del Cargo

dd/mm/aaaa

Debe estar dentro del rango del período seleccionado. Dejar vacío si el cargo está vigente.

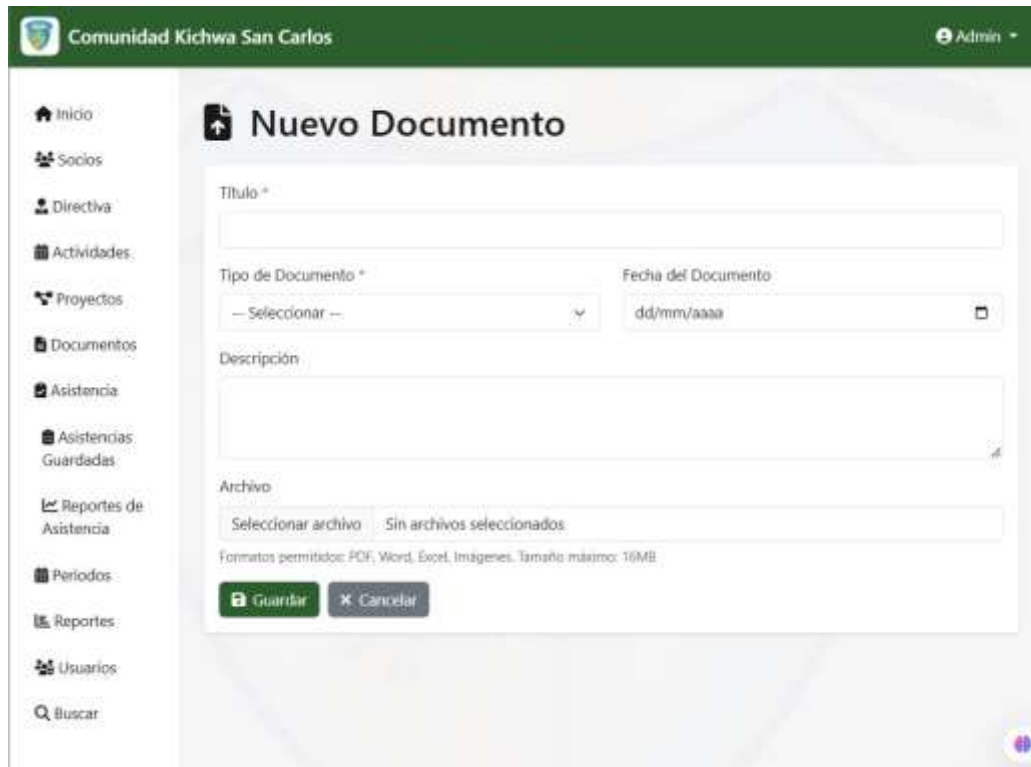
Guardar Cancelar

Nota: La figura muestra el Módulo de Gestión de Directiva, utilizado para registrar cargos directivos y asignarlos a un período institucional.

RF-04: Módulo de Gestión de Documentos

- Subir documentos oficiales
- Categorizar documentos (actas, oficios, resoluciones, informes)
- Consultar y descargar documentos
- Eliminar documentos obsoletos

Figura 15: Módulo de Gestión de Documentos

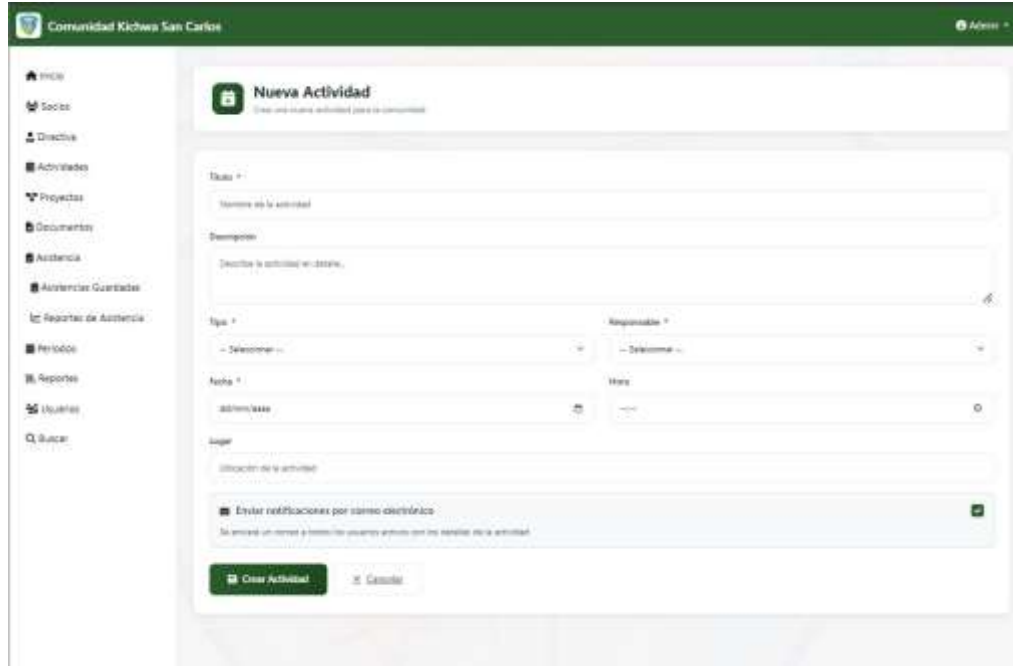


Nota: La figura muestra el Módulo de Gestión de Documentos, que permite registrar, clasificar y almacenar documentos digitales dentro del sistema.

RF-05: Módulo de Gestión de Eventos

- Registrar eventos comunitarios (asambleas, mingas, festividades)
- Programar fechas y ubicaciones
- Registrar asistencia
- Consultar historial de eventos

Figura 16: *Módulo de Gestión de Eventos*



Nota 2: La figura presenta el Módulo de Gestión de Eventos, utilizado para registrar y administrar actividades programadas dentro del sistema.

RF-06: Módulo de Reportes

- Generar reporte de miembros activos
- Generar reporte de directiva actual
- Generar estadísticas de asistencia
- Exportar reportes en PDF

Figura 17: Módulo de Reportes

Resultado del diseño de la arquitectura: Objetivo Específico 2

Objetivo: Diseñar la arquitectura del sistema y la base de datos que almacene la información organizacional de la comunidad, estableciendo los módulos funcionales necesarios.

Resultados obtenidos:

1. Arquitectura del sistema:

Se diseñó una arquitectura de tres capas basada en el patrón MVC:

Capa de Presentación (Vista):

- Plantillas HTML con Jinja2
- Framework CSS Bootstrap 5.3
- JavaScript para interactividad
- Diseño responsivo mobile-first

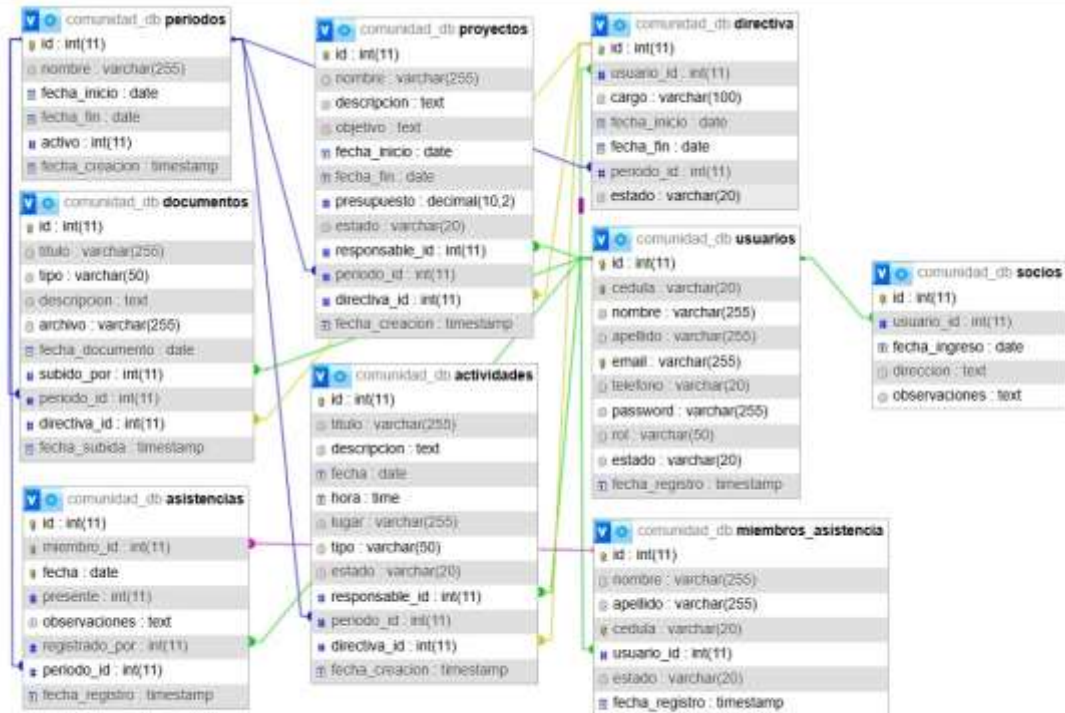
Capa de Lógica de Negocio (Controlador):

- Rutas y controladores Flask
- Validación de formularios con Flask-WTF
- Manejo de sesiones con Flask-Login
- Procesamiento de lógica de negocio **Capa de Datos (Modelo):**
- Modelos ORM con Flask-SQLAlchemy
- Base de datos SQLite
- Gestión de relaciones entre entidades

2. Diseño de la base de datos:

Se diseñó un modelo relacional con las siguientes tablas principales:

Figura 18: *Tabla principal del proyecto*



Nota: La figura presenta el modelo relacional de la base de datos, donde se identifican las tablas principales del sistema y sus relaciones para la correcta gestión de la información.

Resultado del desarrollo del sistema: Objetivo Específico 3

Objetivo: Desarrollar módulos funcionales para el registro, consulta, actualización y eliminación de datos de la comunidad, implementando mecanismos de seguridad que permitan una gestión dinámica y sostenible de la información.

Resultados obtenidos:

1. Sistema web completamente funcional:

Se desarrolló exitosamente el sistema web con todos los módulos planificados, implementado con las siguientes tecnologías:

Figura 19: Panel de Control



Nota: La figura muestra el Panel de Control del sistema, donde se visualiza un resumen general de socios, directivas, actividades y proyectos para facilitar la gestión administrativa.

H. CONCLUSIONES

El análisis de los procesos actuales realizados mediante entrevistas y revisión documental permitió identificar claramente las actividades críticas que la comunidad realiza en la administración de su información. Se determinó que los registros se gestionan de forma manual, dispersa y sin un control centralizado, lo que dificulta la disponibilidad y exactitud de los datos. Esto facilitó la identificación precisa de los requerimientos funcionales necesarios para el diseño del sistema.

El diseño de la arquitectura del sistema y su base de datos permitió estructurar de manera organizada la información de la comunidad, definiendo módulos funcionales acordes a sus necesidades operativas. La arquitectura propuesta garantiza la correcta integración entre los componentes del sistema, asegurando escalabilidad, organización y un manejo eficiente de los datos.

El desarrollo de los módulos funcionales permitió implementar herramientas para el registro, consulta, edición y eliminación de información, incorporando mecanismos de seguridad adecuados. Esto contribuyó a una gestión dinámica y confiable de los datos de la comunidad, mejorando notablemente el control interno y reduciendo errores y pérdidas de información.

I. RECOMENDACIONES

Se recomienda continuar actualizando y documentando periódicamente los procesos administrativos de la comunidad, a fin de mantener vigente la identificación de necesidades y asegurar que el sistema se adapte a cambios futuros en la gestión de la información.

Es aconsejable evaluar y mejorar la arquitectura del sistema conforme aumente el volumen de datos o surjan nuevas necesidades, garantizando que la estructura tecnológica siga siendo eficiente, escalable y alineada a los estándares actuales de desarrollo de software.

Se recomienda fortalecer los mecanismos de seguridad mediante copias de seguridad frecuentes, control de accesos por roles y capacitación a los usuarios del sistema, asegurando una gestión responsable, segura y sostenible de la información comunitaria.

J. BIBLIOGRAFÍA

Asamblea Nacional del Ecuador. (2008). *Constitución de la República del Ecuador*.

Registro Oficial.

Asamblea Nacional del Ecuador. (2015). *Ley Orgánica de Telecomunicaciones*.

Registro Oficial.

Asamblea Nacional del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales*. Registro Oficial.

- Bootstrap Team. (2023). *Bootstrap Documentation*. <https://getbootstrap.com/docs/>
- Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. Wiley.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. [Tesis doctoral, University of California, Irvine].
- Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- Flask Documentation. (2023). *Welcome to Flask*. <https://flask.palletsprojects.com/>
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.
- Instituto Ecuatoriano de Normalización. (2018). *NTE INEN-ISO/IEC 27001: Tecnología de la información — Técnicas de seguridad — Sistemas de gestión de la seguridad de la información — Requisitos*.
- Lutz, M. (2019). *Learning Python* (5th ed.). O'Reilly Media.
- Nielsen Norman Group. (2020). *User Experience: Definition and Overview*. <https://www.nngroup.com/topic/user-experience/>
- Obsbusiness.school. (2020). *Metodologías de desarrollo de software*. <https://obsbusiness.school/>
- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill.
- Sánchez, J. (2016). *Diseño Centrado en el Usuario (UCD)*. [Artículo de blog o publicación interna]. Nota: Como no se encontró la fuente original, se cita como se hace en el documento. Se recomienda buscar una fuente académica más robusta para la versión final.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. <https://scrumguides.org/>
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2014). *Database System Concepts* (6th ed.). McGraw-Hill.
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.
- SQLite Team. (2023). *About SQLite*. <https://www.sqlite.org/about.html>

K. ANEXOS

Anexo 1.- Fotografías

Fotografía 1:

Primer acercamiento con la comunidad



Nota: La fotografía muestra el primer acercamiento con la comunidad, realizado como parte del proceso de levantamiento de información y socialización del proyecto.

Anexo 3.-

Solicitud de autorización

Puerto Napo, 13 de octubre de 2025

Sr.
Carlos Francisco Tapuy Grefa
PRESIDENTE DE LA COMUNIDAD KICHWA SAN CARLOS

Presente

De mi consideración:

Yo, **Medardo Pedro Tapuy Andy**, con C.I. 150087677-4, estudiante de la carrera de Desarrollo de Software del Instituto Superior Tecnológico Tena, me dirijo respetuosamente a usted con la finalidad de solicitar su autorización para el uso del nombre de la *Comunidad Kichwa San Carlos*, así como el acceso a la información de la comunidad, exclusivamente con fines académicos.

La información solicitada será utilizada para el desarrollo del proyecto de titulación titulado "*Desarrollo de un sistema web para la administración de información organizacional de la Comunidad Kichwa San Carlos*", el cual se ejecutará como requisito previo a la obtención del título de Tecnólogo Superior en Desarrollo de Software, encontrándonos actualmente en el último semestre de la carrera.

Agradezco de antemano la atención brindada a la presente solicitud y expreso mi compromiso de utilizar la información proporcionada de manera ética, responsable y conforme a los fines académicos establecidos.

Sin otro particular, me suscribo reiterando mi sincero agradecimiento y deseándole éxitos en sus actividades profesionales.

Atentamente,



Medardo Pedro Tapuy Andy
Estudiante – Instituto Superior Tecnológico Tena
C.I. 150087677-4
Correo electrónico: medardo.tapuy@est.itstena.edu.ec

Recibido
Lunes, 13 de Octubre de 2025
Hora, 12:00

[Firma]
1500277338

COMUNIDAD KICHWA SAN CARLOS
Km. 8 Vía Anuano - Pto. Napo
Ac. Min. 0177 - Filial a FUCONA
PRESIDENCIA
PTO. NAPO - TENA - NAPO - ECUADOR

Anexo 4.-

Solicitud de Aprobación.



COMUNIDAD KICHWA "SAN CARLOS"

DIR.: KM 8 VÍA AHUANO
Acuerdo Ministerial N° 0177 de 17 de Julio de 2019 – SNGP –
Filial a la Federación FUCONA
PTO. NAPO – TENA - ECUADOR

San Carlos, 17 de octubre de 2025

Sr.
Medardo Pedro Tapuy Andy
ESTUDIANTES DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA

Presente

De mi consideración:

Todos los que conformamos la comunidad Kichwa San Carlos, hacemos llegar un cordial saludo al estudiante del proyecto de titulación del Instituto Superior Tecnológico Tena.

En atención a la solicitud de fecha 13 de octubre de 2025, **autorizo** el uso del nombre de la Comunidad Kichwa San Carlos y, a su vez, junto con la información necesaria de la comunidad para la realización del proyecto de titulación titulado: *"Desarrollo de un sistema web para la administración de información organizacional de la Comunidad Kichwa San Carlos"* al estudiante **Medardo Pedro Tapuy Andy** (C.I. 150087677-4).

Por la favorable atención que se digne dar a los presentes, reitero mi sentimiento de consideración y estima.

Atentamente,

Sr. Carlos Francisco Tapuy Grefa
PRESIDENTE DE LA COMUNIDAD KICHWA SAN CARLOS



DIR.: KM 8 VÍA AHUANO, PUERTO NAPO SAN CARLOS – PTO. NAPO - TENA – ECUADOR
E-MAIL: comunidadkichwasanCarlos@gmail.com - CELULAR: 096 952 6688

Anexo 5.-

Certificado de Implementación- Medardo Tapuy



COMUNIDAD KICHWA "SAN CARLOS"

DIR.: KM 8 VÍA AHUANO
Acuerdo Ministerial N° 0177 de 17 de Julio de 2019 – SNGP –
Filial a la Federación FUCONA
PTO. NAPO – TENA - ECUADOR

LA COMUNIDAD KICHWA SAN CARLOS

CERTIFICA:

Se certifica que el señor **Medardo Pedro Tapuy Andy**, con cédula número 1500876774, estudiante del cuarto semestre de la carrera de *Desarrollo de Software* del *Instituto Superior Tecnológico Tena*, realizó la implementación del proyecto de integración curricular en la *Comunidad Kichwa San Carlos* durante el período lectivo octubre de 2025 – febrero de 2026.

El presente certificado podrá ser utilizado en lo que el portador estime necesario, excepto para trámites judiciales.

San Carlos, 4 de febrero de 2026

Sr. Carlos Francisco Tapuy Grefa
PRESIDENTE DE LA COMUNIDAD KICHWA SAN CARLOS

COMUNIDAD KICHWA SAN CARLOS
Km. 8 Vía Ahuano - Pto. Napo
Ac. Min. 0177 - Filial a FUCONA
PRESIDENCIA
PTO. NAPO - TENA - NAPO - ECUADOR

DIR.: KM 8 VÍA AHUANO, PUERTO NAPO SAN CARLOS – PTO. NAPO - TENA – ECUADOR
E-MAIL: comunidadkichwasancarlos@gmail.com - CELULAR: 096 952 6688