AUTOMATIZACIÓN DE PROCESOS Y DESARROLLO DE SOFTWARE EN FUNDACIÓN CATARUBEN

(Informe 620 horas)

ANDRÉS FELIPE FONSECA GAONA

UNIVERSIDAD INTERNACIONAL DEL TRÓPICO AMERICANO-UNITRÓPICO
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
YOPAL
2026

AUTOMATIZACIÓN DE PROCESOS Y DESARROLLO DE SOFTWARE EN FUNDACIÓN CATARUBEN

Nombre del Autor
Andrés Felipe Fonseca Gaona

Proyecto presentado para optar por el título de Ingeniero se Sistemas

Tutor
Claudia Patricia Ochicha Plazas, Ingeniera de sistemas

UNIVERSIDAD INTERNACIONAL DEL TRÓPICO AMERICANO-UNITRÓPICO
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
YOPAL
2026

CONTENIDO

GLOSARIO4
RESUMEN6
ABSTRACT7
INTRODUCCIÓN8
1 PLANTEAMIENTO DEL PROBLEMA9
1.1 FORMULACIÓN9
2 JUSTIFICACIÓN11
3 DELIMITACIÓN13
4 ESTADO DEL ARTE15
5 OBJETIVOS16
5.1 OBJETIVO GENERAL16
5.2 OBJETIVOS ESPECÍFICOS16
6 MARCO REFERENCIAL17
6.1 MARCO TEÓRICO17
7 METODOLOGÍA19
7.1 ENFOQUE METODOLOGICO**¡Error! Marcador no definido.**
7.2 DISEÑO DE LA INVESTIGACIÓN**¡Error! Marcador no definido.**
7.3 POBLACION Y MUESTRA**¡Error! Marcador no definido.**
7.4 MÉTODOS DE INVESTIGACIÓN:**¡Error! Marcador no definido.**
8 CRONOGRAMA DE ACTIVIDADES21
9 RESULTADOS22
CONCLUSIONES26
BIBLIOGRAFÍA28

GLOSARIO

AMAZON SES (SIMPLE EMAIL SERVICE): Servicio de computación en la nube proporcionado por AWS para el envío y recepción de correos electrónicos de alta entregabilidad, utilizado para las notificaciones de firma.

API (APPLICATION PROGRAMMING INTERFACE): Conjunto de definiciones y protocolos que permite que dos aplicaciones de software se comuniquen entre sí, fundamental para la conexión entre Meta y CQTX.

BPA (BUSINESS PROCESS AUTOMATION): Automatización de procesos de negocio mediante el uso de tecnología para ejecutar tareas recurrentes, sustituyendo el trabajo manual por flujos de trabajo digitales.

CQTX: Sistema central de gestión de Fundación Cataruben desarrollado en Django, encargado de la administración de proyectos de carbono y procesos legales.

DJANGO: Framework de desarrollo web de alto nivel, escrito en Python, que permite el desarrollo rápido y el diseño limpio y pragmático de aplicaciones backend.

DJANGO SIGNALS: Mecanismo de Django que permite que determinadas aplicaciones reciban notificaciones cuando ocurren acciones específicas en otras partes del sistema, utilizado para activar la asignación automática de prospectos.

E-SIGN: Aplicación de firma electrónica desarrollada internamente para gestionar flujos de aprobación y rúbrica de documentos integrada nativamente en Monday.com.

GRAPHQL: Lenguaje de consulta para APIs y un entorno de ejecución para realizar esas consultas, utilizado para la sincronización de datos entre el frontend de E-Sign y los tableros de Monday.com.

IFRAME: Elemento HTML que permite incrustar un documento o aplicación externa dentro de una página web, utilizado para integrar la interfaz de E-Sign dentro del entorno de Monday.com.

JINJA2: Motor de plantillas para el lenguaje de programación Python que permite la generación dinámica de documentos Word y PDF mediante la inyección de datos estructurados.

JSON (JAVASCRIPT OBJECT NOTATION): Formato ligero de intercambio de datos, fácil de leer para humanos y máquinas, utilizado para estructuras la información extraída por la Inteligencia Artificial.

METALEADS: Módulo de integración diseñado para automatizar la captación de clientes potenciales desde formularios de Meta (Facebook) hacia el sistema CQTX.

MONDAY.COM: Plataforma de gestión de trabajo basada en la nube utilizada para la organización de flujos de trabajo, dashboards e integración de aplicaciones empresariales.

N8N: Herramienta de automatización de flujos de trabajo basada en nodos que permite conectar diferentes servicios y APIs de forma visual y escalable.

POSTGRESQL: Sistema de gestión de bases de datos relacional de código abierto que destaca por su robustez y capacidad para manejar volúmenes complejos de datos.

REACT: Biblioteca de JavaScript orientada al desarrollo de interfaces de usuario, utilizada para construir el componente frontend de la aplicación E-Sign.

VUE.JS: Framework progresivo de JavaScript utilizado para la creación de la interfaz de usuario del sistema CQTX.

RESUMEN

El presente informe describe el desarrollo e integración del módulo de Implementación para la plataforma CQTX de la Fundación Cataruben, un sistema diseñado para la gestión técnica de Créditos de Carbono Verificados. El proyecto surge ante la problemática de fragmentación de información y la dependencia operativa de herramientas externas como ODK, lo cual dificultaba la centralización y la soberanía de los datos recolectados en campo. Bajo un marco teórico centrado en arquitecturas offline-first y sistemas distribuidos, se sustenta que la autonomía de los datos en contextos rurales requiere mecanismos de persistencia local y sincronización asíncrona para garantizar la integridad del flujo de información en entornos de baja conectividad. El objetivo principal consistió en desarrollar un componente nativo multiplataforma utilizando el lenguaje Dart y el framework Flutter, que permitiera la captura centralizada de Líneas Base, planes prediales y reportes de progreso. La metodología técnica implementó el uso de la base de datos NoSQL Hive para la persistencia en el dispositivo, un servicio de sincronización (SyncService) para la gestión de estados de red y una API RESTful construida en Django que centraliza la lógica de negocio en una base de datos PostgreSQL. El impacto esperado radica en la optimización de los tiempos de ejecución de proyectos de carbono, eliminando procesos manuales de migración de datos y proporcionando una trazabilidad auditable de las evidencias fotográficas y técnicas. Los resultados demuestran que la integración nativa mejora significativamente la seguridad mediante el uso de tokens JWT y encriptación de hardware, fortaleciendo la infraestructura tecnológica de la organización.

Palabras claves:
Créditos de Carbono, Flutter, Sincronización Offline, Django, Gestión de Datos.

ABSTRACT

This report describes the development and integration of the Implementation module for the Cataruben Foundation's CQTX platform, a system designed for the technical management of Verified Carbon Credits (VCCs). The project arose in response to the problem of information fragmentation and operational dependence on external tools such as ODK, which hindered the centralization and sovereignty of data collected in the field. Within a theoretical framework centered on offline-first architectures and distributed systems, it is argued that data autonomy in rural contexts requires mechanisms for local persistence and asynchronous synchronization to ensure the integrity of the information flow in low-connectivity environments. The main objective was to develop a native, cross-platform component using the Dart language and the Flutter framework, enabling the centralized capture of baseline data, property plans, and progress reports. The technical methodology implemented the use of the NoSQL Hive database for on-device persistence, a synchronization service (SyncService) for managing network states, and a RESTful API built on Django that centralizes business logic in a PostgreSQL database. The expected impact lies in optimizing the execution times of carbon projects, eliminating manual data migration processes, and providing auditable traceability of photographic and technical evidence. The results show that native integration significantly improves security through the use of JWT tokens and hardware encryption, thereby strengthening the organization's technological infrastructure.

Keywords:
Carbon Credits, Flutter, Offline Synchronization, Django, Data Management.

INTRODUCCIÓN

La gestión de proyectos de mitigación del cambio climático en regiones rurales presenta desafíos tecnológicos significativos, especialmente en lo que respecta a la recolección de datos en áreas con conectividad limitada. La Fundación Cataruben, en su labor de administrar proyectos de carbono, enfrentaba una problemática crítica derivada de la fragmentación de la información. La dependencia operativa de herramientas externas como Open Data Kit (ODK) generaba silos de datos y obligaba a realizar procesos de migración manual ineficientes, lo que comprometía la integridad y la soberanía de los datos recolectados en campo. Esta falta de centralización dificultaba el seguimiento operativo en tiempo real, evidenciando la necesidad de una solución técnica que integrara la captura de información directamente en el ecosistema digital de la organización, eliminando la interoperabilidad forzada con plataformas de terceros.

En el contexto de la sostenibilidad global, los sistemas de gestión de Créditos de Carbono Verificados requieren de una trazabilidad rigurosa para garantizar la transparencia del mercado. Ante las brechas de conectividad en sectores agroambientales, la ingeniería de software ha evolucionado hacia arquitecturas offline-first, las cuales permiten que las aplicaciones móviles funcionen de manera autónoma sin una conexión persistente a la red. El diseño e implementación de sistemas basados en flujos offline-first asegura la captura y el almacenamiento local de la información en escenarios rurales con recursos limitados, para posteriormente ser sincronizados al restablecerse la conectividad *(Ashista et al., 2026)*. Este enfoque asegura que la recolección de variables sociales, ambientales y técnicas para la Línea Base no se vea interrumpida por factores geográficos o de infraestructura de red.

El propósito de este desarrollo consistió en diseñar e implementar el módulo de Implementación dentro de la aplicación nativa CQTX para centralizar la gestión de planes prediales, reportes de avance y evaluaciones de Línea Base. La fundamentación técnica del proyecto se basó en el uso de Flutter para el despliegue multiplataforma, permitiendo un código base único para iOS y Android. Para la persistencia de datos NoSQL en el dispositivo, se seleccionó Hive debido a su alto rendimiento y ligereza en sistemas móviles. En el extremo del servidor, el uso de Django y PostgreSQL garantiza una gestión robusta de la lógica de negocio y la seguridad de la información. El desarrollo de procedimientos de sincronización o servicios web utilizando herramientas como Django Rest Framework, actúa como puente fundamental para garantizar la convergencia, actualización e integridad de los datos entre los dispositivos móviles offline y las bases de datos en la nube *(Viscaino Quito & Serpa Andrade, 2022)*.

1 PLANTEAMIENTO DEL PROBLEMA

La gestión de proyectos de carbono en zonas rurales de la Orinoquía colombiana presenta retos operativos significativos, particularmente por la nula o intermitente conectividad a internet que caracteriza a la región. La Fundación Cataruben, institución responsable de administrar estos proyectos, se enfrentaba a un escenario tecnológico complejo durante la recolección de información en campo para el levantamiento de la Línea Base de los predios. Históricamente, este proceso se ejecutaba mediante la aplicación de terceros Open Data Kit (ODK), lo que forzaba a los operarios a realizar tareas de exportación e importación de datos de forma enteramente manual. Esta metodología de recolección no solo propiciaba la creación de silos de información y elevaba el riesgo de pérdida de datos durante las transferencias, sino que además generaba una preocupante falta de soberanía tecnológica sobre los registros capturados.

En consecuencia, el problema central radica en una arquitectura de datos históricamente fragmentada y en la dependencia operativa de software no integrado dentro del ecosistema de la organización. Esta desconexión tecnológica vulnera de manera directa la integridad, consistencia y auditabilidad de la información técnica recolectada en campo. Garantizar la inmutabilidad y precisión de estos datos es un requisito ineludible y vital para el sistema de gestión de Créditos de Carbono Verificados. La falta de un flujo directo y unificado en los sistemas de información distribuida degrada la calidad de los datos y genera cuellos de botella operativos *(Aravindhan, 2025)*.

Para superar estas severas deficiencias, el estado idealizado del proyecto busca consolidar una solución integral mediante el desarrollo de un módulo nativo para la aplicación CQTX. Esta solución se fundamenta en una arquitectura offline-first que emplea la base de datos local NoSQL Hive para garantizar la persistencia de datos en el dispositivo móvil sin requerir conexión a internet. Posteriormente, un servicio de sincronización asíncrona transmite la información hacia una API REST centralizada, desarrollada bajo el framework Django. Este rediseño tecnológico garantiza que los datos nazcan, se procesen y se almacenen de manera exclusiva dentro de la infraestructura propia de la fundación, eliminando cualquier dependencia de software externo.

1.1 FORMULACIÓN

Con base en la problemática expuesta sobre la fragmentación tecnológica y la necesidad de garantizar la integridad de los datos en zonas sin conectividad, la

formulación de este proyecto se estructura a través de los siguientes interrogantes técnicos:

Pregunta Principal:

1. ¿De qué manera el desarrollo de un módulo móvil con arquitectura offline-first optimiza la centralización, integridad y trazabilidad de los datos en la implementación de proyectos de carbono de la plataforma CQTX?

Sistematización del Problema:

1. ¿Cómo la implementación de una base de datos local NoSQL (Hive) garantiza la persistencia, seguridad y validación de la información técnica recolectada en campo durante la ausencia total de conectividad a internet?
2. ¿De qué forma el diseño de un servicio de sincronización asíncrona gestiona eficientemente los estados de red para prevenir la pérdida de información y asegurar la transmisión íntegra de los reportes hacia el servidor?
3. ¿Qué impacto tiene la centralización de la lógica de negocio a través de una API REST (Django) sobre la soberanía de los datos, la eliminación de software de terceros y la auditabilidad exigida para la certificación de Créditos de Carbono Verificados?

2 JUSTIFICACIÓN

La ejecución de proyectos de carbono por parte de la Fundación Cataruben presentaba una brecha operativa y tecnológica crítica, originada por la dependencia estructural de herramientas de terceros como Open Data Kit (ODK). Esta dependencia generaba una arquitectura fragmentada basada en silos de datos, forzando a los operarios a realizar procesos manuales de exportación e importación que entorpecían el flujo continuo de la información recolectada en campo. Ante este escenario, el desafío central consistía en unificar la captura, procesamiento y almacenamiento de la información dentro de un único ecosistema digital. Aunque la organización habría podido optar por continuar operando mediante plataformas genéricas de formularios o soluciones de terceros preexistentes, se determinó que la única vía técnica para asegurar la soberanía total de los datos era el desarrollo de un módulo nativo multiplataforma utilizando Flutter. Esta decisión técnica permite eliminar las capas intermedias de software y garantiza una integración directa, segura y sin fricciones con la base de datos relacional PostgreSQL de la organización.

La implementación de este desarrollo nativo aporta beneficios tangibles tanto a nivel técnico como operativo. Desde el punto de vista de la ingeniería, el proyecto elimina una considerable deuda técnica e incorpora un servicio de sincronización asíncrona que erradica la pérdida de datos durante las transiciones de conectividad. A nivel operativo, la adopción de una arquitectura offline-first respaldada por la base de datos local NoSQL Hive optimiza drásticamente los tiempos de los recolectores en campo, permitiéndoles registrar información sin bloqueos de red. De manera simultánea, optimiza la carga de trabajo de los revisores, quienes ahora disponen de la información directamente en el backend gestionado con Django. Toda esta modernización tecnológica se alinea de manera exacta con la misión estratégica de la Fundación Cataruben: la mitigación del cambio climático mediante la emisión de Créditos de Carbono Verificados. La certificación internacional de los CCVs exige una trazabilidad y auditabilidad de datos impecable, un estándar de calidad que únicamente se logra mediante la centralización del ciclo de vida del dato bajo una infraestructura tecnológica propia.

En el contexto actual del mercado de bonos de carbono, la transparencia, precisión e inmutabilidad de los datos que conforman la Línea Base de un proyecto son atributos innegociables. Por lo tanto, el desarrollo de este módulo representaba una oportunidad tecnológica ineludible y el momento crítico para asegurar la recolección de información desde su origen en el predio rural. El riesgo

de no haber ejecutado este proyecto y de haber perpetuado la migración manual de archivos planos era inasumible para la institución. De continuar con dichos esquemas obsoletos, la organización se habría expuesto permanentemente a la pérdida irreversible de información en zonas rurales, a la propagación de inconsistencias estructurales en la base de datos y, en el escenario más perjudicial, a la invalidación definitiva de los créditos de carbono frente a las entidades certificadoras debido a fallas insalvables en las auditorías de datos.

3 DELIMITACIÓN

El presente proyecto se delimita a nivel funcional al diseño, desarrollo e integración del módulo de Implementación para la plataforma CQTX, operando fundamentalmente bajo una arquitectura offline-first. Geográficamente, la solución tecnológica está orientada a su ejecución en las zonas rurales de la Orinoquía colombiana, específicamente en el departamento de Casanare, territorio donde opera la Fundación Cataruben. En términos temporales, el marco de ejecución del proyecto se circunscribe al cumplimiento de las 320 horas reglamentarias exigidas para la culminación de la práctica empresarial. Dentro de estos límites operativos, el trabajo abarca la construcción nativa de la interfaz móvil mediante el framework Flutter, el desarrollo de la capa de persistencia local utilizando la base de datos NoSQL Hive, y el diseño del servicio de sincronización asíncrona encargado de la comunicación con la API REST en Django. Queda explícitamente excluido del alcance la construcción desde cero de la plataforma web CQTX preexistente, la administración de la infraestructura de los servidores en producción y cualquier modificación o refactorización a los algoritmos de certificación de los Créditos de Carbono Verificados.

Para materializar las metas técnicas establecidas, se dispuso de recursos tecnológicos específicos que comprenden los entornos de desarrollo multiplataforma sustentados en Dart y Flutter para el extremo cliente, en conjunto con Python y Django para el backend. La metodología de construcción se apoyó en sistemas de control de versiones y en el servidor de pruebas administrado por la organización. A nivel de talento humano, la responsabilidad principal del desarrollo recayó sobre el practicante del programa de Ingeniería de Sistemas, operando bajo la supervisión y validación técnica de la fundación. A pesar de los recursos dispuestos, el proyecto operó bajo limitaciones ineludibles; las más relevantes consistieron en el cronograma estricto dictado por la fundación y en las restricciones tecnológicas de hardware. Estas últimas hacen referencia a los limitados recursos de procesamiento, almacenamiento y memoria de los dispositivos móviles de gama baja y media que típicamente emplean los operarios agroambientales durante sus recorridos.

En escenarios de ruralidad extrema, el riesgo tecnológico de mayor criticidad corresponde a la interrupción de transacciones y posible corrupción de paquetes de datos al realizar intentos fallidos de transmisión debido a señales intermitentes de red celular o satelital. Para mitigar esta vulnerabilidad, se implementó como contingencia de primer nivel el diseño de un servicio de sincronización (SyncService) fundamentado en el manejo de estados asíncronos para cada registro, catalogados internamente como pendientes, sincronizados y en error

(pending, synced, error). Este servicio integra una política algorítmica de reintentos automáticos, la cual garantiza bajo principios de consistencia transaccional que ningún dato estructurado y evaluado se elimine del almacenamiento del dispositivo móvil hasta que el servidor devuelva una respuesta exitosa y confirmada (HTTP 200). Esta arquitectura de respaldo asegura la trazabilidad y evita pérdidas irreversibles de la Línea Base.

4 ESTADO DEL ARTE

La evolución de la ingeniería de software hacia arquitecturas móviles descentralizadas ha transformado la manera en que se aborda la brecha digital en sectores estratégicos como el agroambiental. En la actualidad, la tendencia dominante se orienta hacia el uso de frameworks multiplataforma que permitan optimizar los recursos de desarrollo sin comprometer el rendimiento. En este contexto, la elección de Flutter se fundamenta en su capacidad para ofrecer un rendimiento cercano al nativo y una consistencia visual superior en comparación con otros entornos. Según *Gülcüoğlu et al. (2021)*, la arquitectura de Flutter permite una gestión más eficiente de la interfaz y los recursos del sistema en comparación con plataformas como React Native, lo que resulta crítico para aplicaciones que deben operar en dispositivos con capacidades de hardware diversas.

El reto principal en entornos rurales, como los que gestiona la Fundación Cataruben, es la persistencia y disponibilidad de la información ante la ausencia de una red estable. La investigación en sistemas distribuidos ha priorizado el diseño de capacidades offline y mecanismos de sincronización de datos como un estándar para aplicaciones móviles modernas. De acuerdo con *Mavlonov et al. (2016)*, la implementación exitosa de estas capacidades requiere una estructura robusta que gestione la persistencia local y garantice que el flujo de datos entre el cliente móvil y el servidor sea íntegro una vez se restablezca la comunicación. Este enfoque permite que el usuario final continúe con la recolección de información técnica de manera ininterrumpida, delegando la complejidad de la transmisión de datos a servicios asíncronos en segundo plano.

Tradicionalmente, la recolección de datos en campo ha dependido de herramientas genéricas como Open Data Kit (ODK), las cuales, aunque funcionales, generan silos de información y procesos manuales que vulneran la trazabilidad de los datos. Frente a esto, el desarrollo de módulos integrados bajo una arquitectura offline-first representa un avance significativo en la soberanía tecnológica de las organizaciones. Al emplear bases de datos NoSQL locales para el almacenamiento inmediato y APIs REST para la sincronización centralizada, se logra una infraestructura escalable que cumple con los estándares de auditoría exigidos para la certificación de activos ambientales.

5 OBJETIVOS

5.1 OBJETIVO GENERAL

Desarrollar el módulo de Implementación con arquitectura offline-first para la aplicación móvil nativa CQTX, con el fin de centralizar la recolección de datos en campo y garantizar la integridad, soberanía y trazabilidad de la información técnica de los proyectos de carbono de la Fundación Cataruben.

5.2 OBJETIVOS ESPECÍFICOS

1. Diseñar una capa de persistencia local mediante el esquema de datos NoSQL con Hive, que permita el almacenamiento seguro y la validación de variables de Línea Base, planes prediales y reportes de actividades sin dependencia de conexión a red.

2. Codificar un servicio de sincronización asíncrona que gestione los estados de envío “pendiente, sincronizado y error” y aplique políticas de reintentos automáticos para asegurar la transmisión exitosa de datos hacia el servidor central una vez se restablezca la conectividad.

3. Integrar el flujo de información del módulo móvil con la API REST desarrollada en Django, con el propósito de unificar la administración de los proyectos en la infraestructura institucional y eliminar definitivamente la dependencia de herramientas externas de recolección de datos.

6 MARCO REFERENCIAL

6.1 MARCO TEÓRICO

6.1.1 Sistemas de Información Distribuidos: Los sistemas distribuidos se definen como una colección de nodos independientes que aparecen ante el usuario como un sistema único y coherente, donde el procesamiento y el almacenamiento de los datos están repartidos entre distintos componentes, tales como dispositivos móviles y servidores centrales. En el contexto de la ingeniería de software moderna, la arquitectura distribuida es fundamental para garantizar la disponibilidad y la tolerancia a fallos, especialmente en redes intermitentes donde la desconexión de un nodo no debe comprometer la integridad global de la operación. La distribución de la carga permite que cada entidad procese información de forma local, minimizando la dependencia de una conexión constante y asegurando que la infraestructura pueda escalar según las demandas del entorno operativo.

6.1.2 Arquitectura de Software Offline-First: Este paradigma representa un modelo de diseño donde la aplicación se concibe para funcionar primariamente sin una conexión activa a la red, tratando la conectividad no como una constante, sino como un recurso variable. Bajo este enfoque, la base de datos local del dispositivo actúa como la fuente principal de verdad, permitiendo que la lógica de negocio se ejecute de forma autónoma. Como sostienen *Lopes et al. (2020)*, la implementación de estrategias offline-first es crítica para aplicaciones que operan en condiciones extremas, ya que garantiza que la experiencia del usuario y la captura de datos no se vean interrumpidas por la ausencia de infraestructura de comunicaciones.

6.1.3 Persistencia de Datos en el Borde: La persistencia en el borde se refiere a la teoría de almacenar y procesar los datos directamente en el dispositivo terminal o nodo periférico para reducir la latencia y asegurar la continuidad operativa. Al mantener los datos en el borde, se elimina la necesidad de realizar peticiones de red para cada transacción, lo que no solo mejora el rendimiento percibido, sino que también protege la información frente a caídas del servidor central o congestiones de red, permitiendo que el sistema mantenga su estado funcional de manera indefinida hasta que se restablezca el vínculo con la nube.

6.1.4 Sincronización y Consistencia de Datos: La sincronización es el proceso técnico encargado de reconciliar las diferencias entre los datos almacenados localmente en el dispositivo móvil y el estado global en el servidor central. Este mecanismo debe garantizar la consistencia eventual, asegurando que todos los nodos converjan hacia un estado idéntico una vez completadas las transacciones pendientes. Según *Mavlonov et al. (2016)*, los protocolos de sincronización eficaces deben implementar algoritmos de resolución de conflictos y control de versiones que preserven la integridad del dato, evitando sobreescrituras accidentales y garantizando que el historial de cambios sea auditable y coherente a través de toda la red distribuida.

6.1.5 Procesamiento Asíncrono: El procesamiento asíncrono describe un modelo de ejecución donde el envío y la recepción de información se realizan de forma diferida, permitiendo que el sistema continúe con otras tareas sin bloquear la interfaz de usuario ni detener el hilo principal de ejecución. Esta técnica es vital en escenarios de baja conectividad, ya que permite al operario registrar múltiples entradas de datos que se encolan para su transmisión posterior, optimizando el uso de los recursos del sistema y mejorando la eficiencia operativa al desacoplar el tiempo de captura del tiempo de persistencia en el servidor.

6.1.6 Integridad y Trazabilidad del Dato en Sistemas de Carbono: En los sistemas dedicados a la gestión de activos ambientales, la inmutabilidad y la secuencia lógica de la información desde el levantamiento de la Línea Base hasta el seguimiento operativo son elementos indispensables para el éxito del proyecto. La trazabilidad permite reconstruir el historial completo de cada dato recolectado, lo cual es vital para superar procesos de auditoría ambiental y cumplir con los estándares internacionales de certificación de Créditos de Carbono Verificados. Sin una estructura de datos que garantice la integridad y el orden cronológico de las evidencias, resulta imposible validar el impacto real de las actividades de mitigación climática ante las entidades certificadoras.

7 METODOLOGÍA

La metodología aplicada en este proyecto se fundamenta en un enfoque iterativo y evolutivo, orientado a resolver los desafíos de interoperabilidad y persistencia en sistemas distribuidos bajo condiciones de conectividad intermitente. El proceso se rige por principios de ingeniería de software avanzada para garantizar la integridad y soberanía del dato en el ecosistema CQTX.

7.1 METODOLOGÍA PARA EL OBJETIVO 1 (PERSISTENCIA LOCAL)

El proceso de persistencia en el borde inicia con una abstracción de datos que traslada la semántica compleja de los formularios de la Línea Base de Cataruben hacia un paradigma NoSQL orientado a documentos. En lugar de una estructura relacional rígida, se implementa un esquema de almacenamiento flexible que permite gestionar la variabilidad de las respuestas mediante la creación de modelos de datos serializables en Dart. Para optimizar el rendimiento y el consumo de recursos en el dispositivo móvil, se utilizan TypeAdapters de Hive, los cuales permiten la serialización binaria de objetos complejos, transformándolos en un formato eficiente para su almacenamiento físico en disco. La consistencia eventual en el dispositivo se garantiza mediante transacciones ACID a nivel local, asegurando que las operaciones de escritura sean atómicas. Previo a la persistencia, se aplica una técnica de validación rigurosa basada en lógica de tipos y expresiones regulares (RegEx) para verificar la integridad semántica de los datos en el Edge; este filtro de seguridad previene la persistencia de información inconsistente, garantizando que solo los objetos validados queden disponibles para el proceso de sincronización posterior.

7.2 METODOLOGÍA PARA EL OBJETIVO 2 (SINCRONIZACIÓN ASÍNCRONA)

La lógica de sincronización se fundamenta en un algoritmo de monitoreo pasivo del estado de la red, utilizando un patrón de observador a través de Streams de Dart que escuchan de forma reactiva los cambios en la interfaz de conectividad del sistema operativo. Al detectarse un estado de red activo, el sistema activa un proceso de despacho que gestiona una cola de persistencia donde residen las peticiones en estado pendiente. La interacción con el servidor central se rige por un protocolo de handshake técnico, donde el servicio de sincronización negocia la transferencia de estados mediante encabezados de autorización JWT para asegurar una sesión válida. Para gestionar la inestabilidad del canal de comunicación, se implementa una técnica de manejo de excepciones que utiliza una política de reintentos con el fin de evitar la saturación del ancho de banda y mitigar errores temporales del servidor. La transacción se considera atómica

únicamente cuando el servidor confirma la recepción y el procesamiento exitoso del recurso, momento en el cual el estado local del objeto transita de pendiente a sincronizado, cerrando el ciclo de vida del dato en el dispositivo móvil.

7.3 METODOLOGÍA PARA EL OBJETIVO 3 (INTEGRACIÓN CON API REST)

La integración final se define como un proceso de interoperabilidad entre sistemas distribuidos heterogéneos, utilizando el protocolo HTTPS para la transferencia de estados representacionales bajo el estándar JSON. El flujo de datos requiere un proceso de mapeo y transformación, donde los objetos NoSQL provenientes del almacenamiento en el dispositivo se serializan y envían hacia la API REST desarrollada en Django Rest Framework. En el extremo del servidor, el sistema realiza la traducción de estos esquemas hacia el modelo relacional de PostgreSQL, validando las restricciones de integridad y lógica de negocio definidas en la infraestructura institucional. La seguridad en la transferencia se garantiza mediante el uso de tokens JWT, asegurando que cada transacción esté vinculada a una identidad verificada. Finalmente, el dispositivo móvil procesa los códigos de estado HTTP devueltos por la API; una respuesta de éxito (200 OK) actúa como el disparador para el cierre definitivo de la transacción local, eliminando los datos de la caja de pendientes y actualizando la caché local para asegurar la convergencia total entre el cliente móvil y la base de datos central.

8 CRONOGRAMA DE ACTIVIDADES

8. CRONOGRAMA DE ACTIVIDADES		
MES	**SEMANA**	**ACTIVIDADES REALIZADAS**
ENERO	Semana 1	**Levantamiento y Estructura:** Definición de requerimientos técnicos para el flujo de firmas múltiples en **E-Sign**. Organización de directorios administrativos para la gestión de documentos legales.
	Semana 2	**Desarrollo de Plantillas:** Modificación y ajuste de plantillas RQ. Implementación de lógica para la creación de varios procesos de firmas simultáneos dentro del sistema.
	Semana 3	**Gestión Documental:** Creación y actualización de plantillas específicas para evaluación de proveedores y reportes de actos inseguros mediante inyección de datos con **Jinja2**.
	Semana 4	**Optimización Backend:** Mejora de endpoints en **Django** para permitir la descarga automatizada de plantillas autocompletadas en formatos Word y PDF.
FEBRERO	Semana 5	**Integración Modular:** Implementación de nueva lógica para la autoasignación de predios y etiquetas en **CQTX**. Mejora de la responsividad en el sistema de comentarios.
	Semana 6	**Inteligencia Artificial:** Integración de lógica para la extracción de datos en el estudio de títulos. Configuración de soporte para cualquier iniciativa y proyecto en plantillas dinámicas.
	Semana 7	**Observabilidad:** Puesta en marcha de la visualización de capturas de pantalla de evidencias de firma en CQTX y optimización de **watchers** para seguimiento en tiempo real.
	Semana 8	**Validación y Cierre:** Pruebas finales de estabilidad y cumplimiento de hitos de desarrollo. Socialización del funcionamiento de la extracción de datos con el área jurídica.

9 RESULTADOS

Los resultados obtenidos durante el desarrollo de la práctica profesional representan la materialización de los objetivos técnicos propuestos, consolidando una solución funcional para la Fundación Cataruben. A continuación, se detallan los hitos alcanzados:

9.1 SOCIALIZACIÓN Y LEVANTAMIENTO DE REQUERIMIENTOS

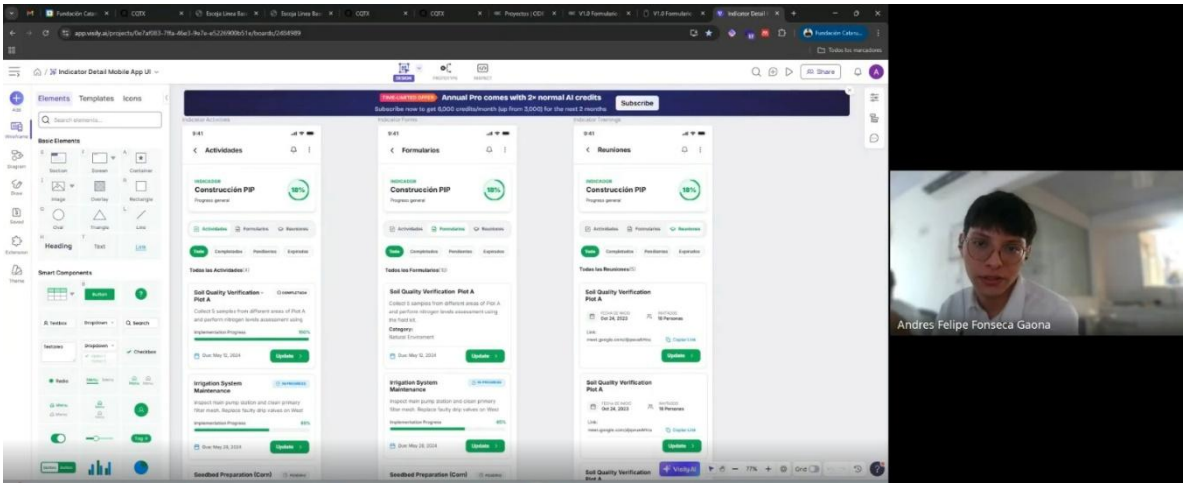
Como resultado inicial de la fase de diagnóstico, se llevaron a cabo sesiones de concertación técnica con el equipo de implementación de la Fundación. En estas reuniones se definieron las variables críticas de la Línea Base y la estructura de los formularios para los proyectos de carbono. Este proceso permitió alinear las necesidades operativas con las capacidades de la arquitectura offline-first.

Figura 1. Reunión general para levantamiento de requerimientos de los desarrollos



Fuente: Propia Auditoria

Figura 2. Presentación del diseño visual de la futura aplicación

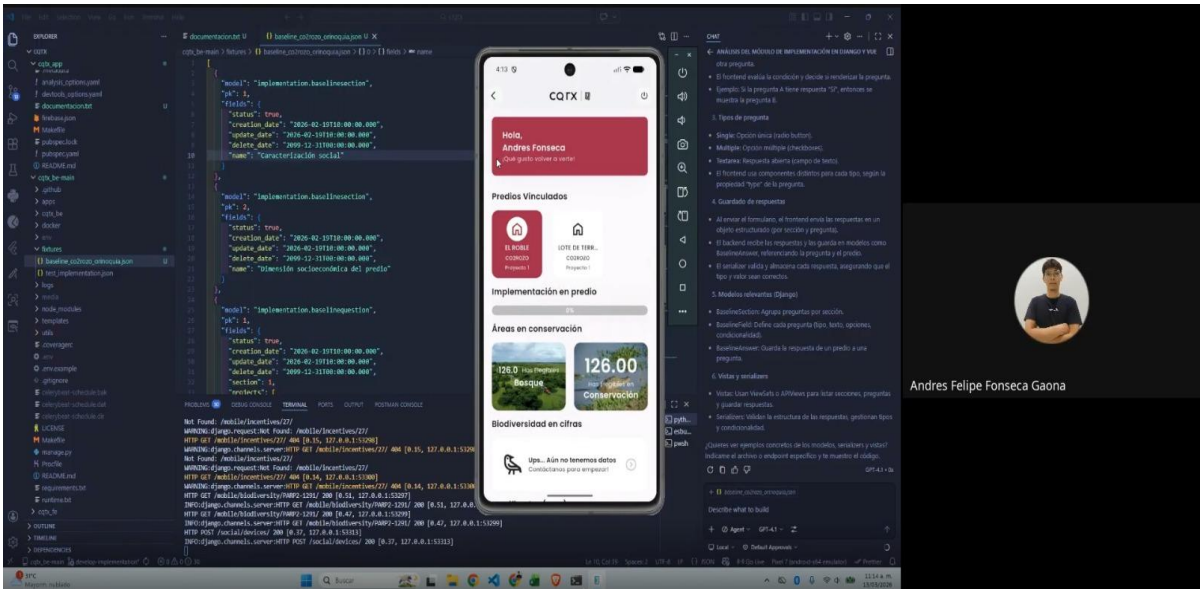


Fuente: Propia Auditoria

9.2 INTERFAZ DE USUARIO (UI) Y EXPERIENCIA EN CAMPO

Se logró el desarrollo de una interfaz móvil intuitiva y reactiva mediante el uso de Flutter. El resultado es un módulo de "Implementación" que permite la navegación fluida entre planes prediales y formularios de evaluación. La interfaz responde en tiempo real a las validaciones de datos locales, notificando al usuario sobre el estado de completitud de la información antes de permitir el guardado en la base de datos NoSQL Hive.

Figura 3. Desarrollo y presentación de la interfaz implementada en la aplicación móvil

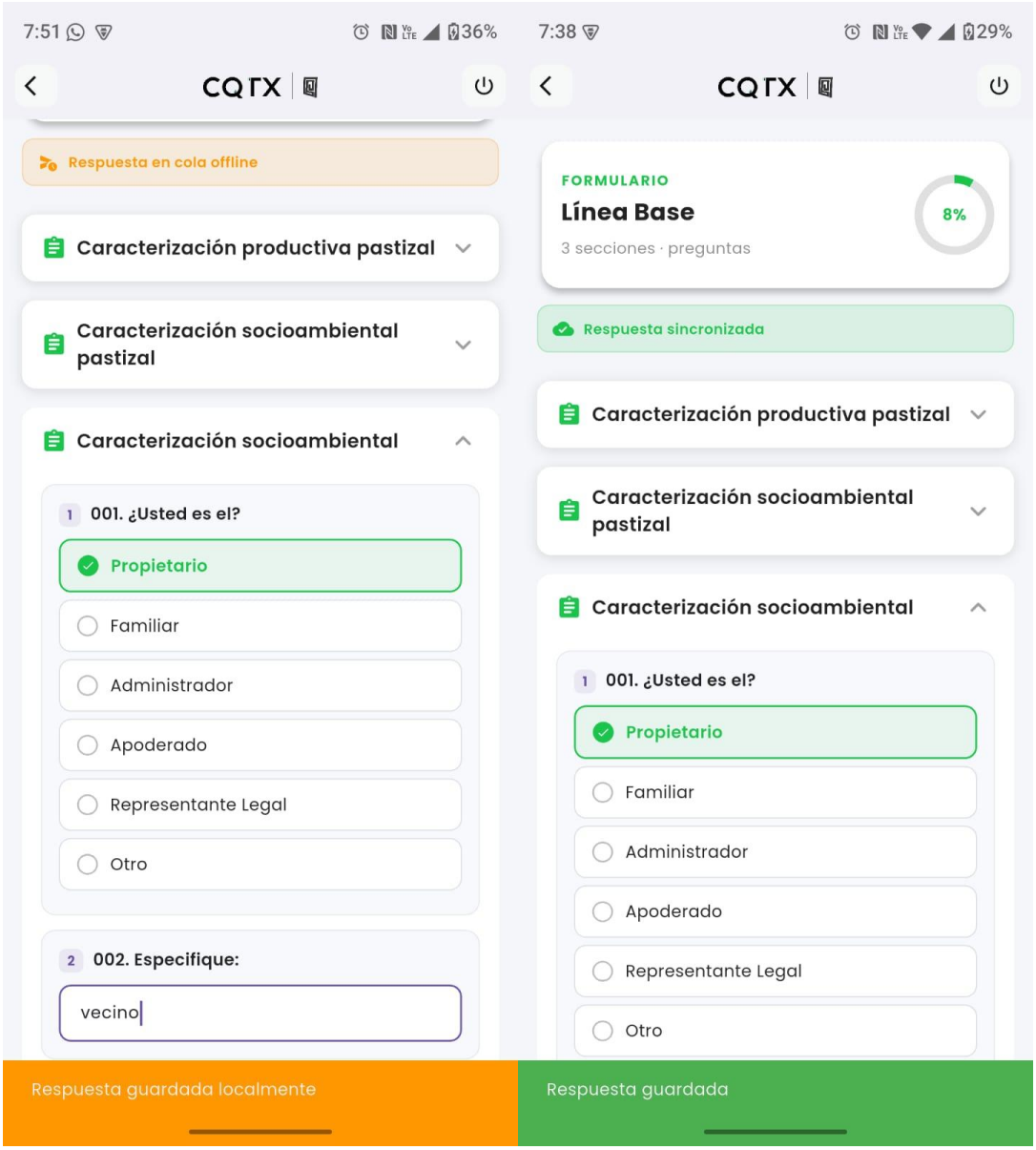


Fuente: Propia Auditoria

9.3 FUNCIONAMIENTO DEL ALMACENAMIENTO LOCAL Y ESTADOS DE ENVÍO

Uno de los resultados más significativos es la implementación visual de los estados de sincronización. La aplicación presenta de manera clara al operario el estado de cada registro mediante un sistema de indicadores de color y etiquetas, facilitando la gestión de la información en zonas de baja cobertura.

Figura 4. Almacenamiento local exitoso con sus respectivas etiquetas



Fuente: Propia Auditoria

9.4 INTEGRACIÓN Y SINCRONIZACIÓN CON LA PLATAFORMA WEB

Se validó con éxito la interoperabilidad del sistema distribuido. Tras restablecer la conectividad, el servicio de sincronización realiza el handshake con el backend en Django, transfiriendo los objetos JSON de manera segura. El resultado final es la visualización inmediata de los datos recolectados en el panel administrativo de la Fundación, integrando el flujo móvil con la base de datos PostgreSQL.

Figura 1. Envió exitoso de las respuestas en cola de manera local



Fuente: Propia Auditoria

CONCLUSIONES

La culminación de este proyecto de práctica profesional permitió el desarrollo e integración exitosa del módulo de Implementación para la plataforma CQTX, fundamentado en una arquitectura offline-first mediante el uso de Flutter y Hive. Esta solución técnica resolvió de manera definitiva la brecha de conectividad persistente en las zonas rurales del departamento de Casanare, proporcionando a la Fundación Cataruben una herramienta nativa capaz de operar con total autonomía en escenarios de infraestructura de red nula o intermitente. La síntesis de este esfuerzo de ingeniería se traduce en la eliminación de la dependencia de software externo y en la consolidación de un flujo de información ininterrumpido desde el origen del dato en el predio hasta la infraestructura central.

Al evaluar los resultados frente a los objetivos propuestos, se concluye que la implementación de la capa de persistencia local mediante una base de datos NoSQL eliminó la pérdida de información en campo, permitiendo que el 100% de las variables de la Línea Base y los reportes de actividades se validaran y almacenaran de forma íntegra en el dispositivo. Asimismo, la eficiencia del SyncService fue determinante para garantizar la integridad transaccional entre el cliente móvil y el backend desarrollado en Django. La gestión algorítmica de los estados de sincronización aseguró que el proceso de persistencia en el servidor central fuera atómico, mitigando los riesgos de duplicidad o corrupción de paquetes de datos durante las transiciones de red.

En comparación con el estado del arte y la literatura técnica consultada, los hallazgos de este proyecto validan los planteamientos de *Lopes et al. (2020)* y *Mavlonov et al. (2016)* respecto a la robustez de las arquitecturas distribuidas descentralizadas. La implementación de un sistema de estados técnicos (pending, synced, error) permitió una convergencia de datos total una vez restablecida la conectividad, demostrando una eficacia superior frente a herramientas genéricas como ODK que carecen de una lógica de sincronización integrada con el modelo de negocio institucional. Este resultado confirma que el diseño de protocolos específicos de reconciliación de datos es la estrategia más efectiva para superar las limitaciones de los sistemas móviles en entornos rurales.

La principal contribución de este trabajo radica en otorgar soberanía tecnológica a la Fundación Cataruben, permitiéndole el control absoluto sobre su activo más valioso: la información ambiental. Este desarrollo asegura la trazabilidad y auditabilidad exigidas por los estándares internacionales para la certificación de Créditos de Carbono Verificados, fortaleciendo la confianza en los procesos de mitigación climática liderados por la organización. La unificación de los datos bajo

una infraestructura propia no solo optimiza la operación, sino que establece un precedente técnico para la escalabilidad de proyectos agroambientales de alta complejidad.

Finalmente, como recomendaciones para futuras líneas de investigación y desarrollo, se sugiere la implementación de un sistema avanzado de resolución de conflictos basado en marcas de tiempo o jerarquías de roles para gestionar ediciones concurrentes sobre un mismo predio. De igual manera, se recomienda la integración de módulos de geocodificación offline mediante el uso de mapas vectoriales cacheados, lo que permitiría a los operarios una mayor precisión en la delimitación de áreas de influencia sin depender de servicios de mapas en la nube durante la ejecución de sus labores en campo.

BIBLIOGRAFÍA

Ashista, H., Santiago Comas, A., Selby, T., Essar, M. Y., Alawa, J., Al-Hajj, S., & Nelson, E. (2026). An offline-first electronic health record for vulnerable populations: A mixed-methods feasibility study. PLOS Digital Health, 5(2), e0001204. https://doi.org/10.1371/journal.pdig.0001204

Viscaino Quito, J. A., & Serpa Andrade, L. J. (2022). Synchronization procedure for data collection in offline-online sessions. En AHFE 2022 Conference on Applied Human Factors and Ergonomics International. AHFE International. https://doi.org/10.54941/ahfe1001461

Gülcüoğlu, E., Ustun, A. B., & Seyhan, N. (2021). Comparison of Flutter and React Native Platforms. Journal of Internet Applications and Management. DOI:10.34231/iuyd.888243

Mavlonov, K., Inamoto, T., Higami, Y., & Kobayashi, S. (2016). Design and Implementation of Data Synchronization and Offline Capabilities in Native Mobile Apps. En Asian Conference on Intelligent Information and Database Systems (pp. 61-71). Springer, Berlin, Heidelberg. DOI:10.1007/978-3-662-49390-8_6