****

**Guía1. Definición Proyecto APT**

**Asignatura Capstone**

1. **PARTE I**

|  |
| --- |
| **1. Antecedentes Personales** |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

|  |  |
| --- | --- |
| Nombre estudiante | **José Almontes – Carlos Navarro – Giovanni Vargas** |
| Rut | **20.845.969-4, 16.104.030-4, 21.230.991-5** |
| Carrera | **Ingeniería en Informática** |
| Sede | **Duoc UC – Sede Viña del Mar** |

|  |
| --- |
| **2. Descripción Proyecto APT** |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

|  |  |
| --- | --- |
| Nombre del proyecto | EcoTrack |
| Área (s) de desempeño(s) | Desarrollo de software, gestión de proyectos informáticos, integración de tecnologías emergentes (IA, Cloud, DevOps). |
| Competencias | Diseñar e implementar soluciones informáticas seguras y escalables.  Gestionar proyectos bajo marcos ágiles y estándares de calidad.  Integrar servicios y arquitecturas orientadas a la nube.  Desarrollar aplicaciones móviles/web con foco en usabilidad y trazabilidad. |

|  |
| --- |
| **3. Fundamentación Proyecto APT** |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. |

|  |  |
| --- | --- |
| Relevancia del proyecto APT | Actualmente, en Cosemar SPA, la gestión de incidentes se realiza de forma manual, dispersa y con baja custodia de evidencias, lo que provoca riesgos legales, dificultades de trazabilidad y tiempos de respuesta elevados. Esta problemática es crítica para el campo laboral de la Ingeniería en Informática, pues exige diseñar soluciones que integren trazabilidad, notificación automática y seguridad de datos.  El proyecto impacta directamente en la región de Valparaíso, en las comunas donde opera Cosemar, beneficiando tanto a la empresa como a sus clientes y usuarios, al asegurar la correcta atención de incidentes y la generación de evidencia confiable.   * El aporte de valor consiste en entregar una plataforma digital integral que simula o implementa en la realidad un sistema de trazabilidad, clasificación automática con IA, reportería con validez legal y alertas en tiempo real, contribuyendo a la modernización tecnológica del sector ambiental. |
| Descripción del Proyecto APT | El proyecto busca desarrollar una **plataforma móvil, web y Backend** que permita:   * Registro de incidentes en terreno con geolocalización, evidencias multimedia y folios únicos. * Integración de IA para clasificación automática de severidad. * Automatización de notificaciones vía WhatsApp y correo electrónico. * Reporterías oficiales en PDF con sello temporal y código QR. * Dashboard web con KPIs en tiempo real y trazabilidad completa. La problemática se abordará mediante un desarrollo iterativo con **Scrum**, entregando valor en cada sprint y validando los entregables con los interesados. |
| Pertinencia del proyecto con el perfil de egreso | El proyecto se alinea con el perfil de egreso de Ingeniería en Informática, ya que exige aplicar competencias de desarrollo full Stack, gestión ágil de proyectos, integración de tecnologías emergentes (IA, Cloud, DevOps) y aseguramiento de la calidad. Resolver la problemática requiere de las competencias de diseño, implementación y gestión de soluciones tecnológicas complejas, todas declaradas en el perfil de la carrera. |
| Relación con los intereses profesionales | Como futuros ingenieros en informática, los integrantes del equipo tienen interés en desempeñarse en áreas como **desarrollo de software, inteligencia artificial, DevOps y ciberseguridad**. Este proyecto se relaciona directamente con dichos intereses, pues permite:   * Practicar la integración de IA para clasificación de datos. * Desarrollar soluciones móviles y web con servicios en la nube. * Aplicar DevOps (CI/CD, monitoreo y seguridad).   Generar experiencia real en proyectos con impacto ambiental y social. Su desarrollo fortalece el perfil profesional del equipo, otorgando experiencia práctica aplicable al mundo laboral. |
| Factibilidad de desarrollo del Proyecto APT | El proyecto es factible porque:   * **Duración:** se planificó en 11 semanas distribuidas en 4 Sprint’s, lo que permite entregas incrementales y controladas. * **Horas asignadas:** la asignatura de Capstone entrega suficiente dedicación para cumplir con las fases del proyecto. * **Materiales:** se cuenta con servicios cloud de bajo costo (Firebase, MongoDB Atlas, OpenAI API, WhatsApp Cloud API). * **Factores facilitadores:** experiencia del equipo en desarrollo web/móvil, uso de Frameworks ágiles y repositorios en GitHub. * **Factores dificultadores:** conectividad en terreno (mitigado con modo offline) y costos de uso de IA (mitigado con Caching y plan académico).   ¿Por qué crees es posible desarrollar tu Proyecto APT? Para responder esta pregunta debemos tener en consideración:   1. Duración del semestre 2. Horas asignadas a la asignatura 3. Materiales requeridos 4. Factores externos que facilitan su desarrollo 5. Factores externos que dificultan su desarrollo y maneras en que podríamos solucionarlos |

1. **PARTE II**

|  |
| --- |
| **4. Objetivos** |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

|  |  |
| --- | --- |
| Objetivo general | Desarrollar una plataforma digital integral (EcoTrack) que permita registrar, clasificar, notificar, auditar y reportar incidentes, logrando reducir en al menos un 60% los tiempos de asignación (TTA) y resolución (TTR) en un plazo de 11 semanas. |
| Objetivos específicos | Implementar una aplicación móvil para el registro de incidentes con geolocalización y evidencias multimedia.  Integrar inteligencia artificial para clasificación automática de incidentes.  Automatizar notificaciones inmediatas vía WhatsApp y correo electrónico.  Generar reportes PDF oficiales con sello temporal y QR de validación.  Construir un Dashboard web con KPIs en tiempo real y trazabilidad auditada. |

|  |
| --- |
| **5. Metodología** |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

|  |
| --- |
| Descripción de la Metodología |
| **Metodología seleccionada:** *Scrum + Product Backlog Building (PBB)*.   * **Etapas:**   + Sprint 1 (Semanas 1–3): Registro base, offline y API/DB.   + Sprint 2 (Semanas 4–5): MVP con IA, evidencias y Dashboard básico.   + Sprint 3 (Semanas 6–8): Notificaciones, reportería PDF y KPIs.   + Sprint 4 (Semanas 9–11): RBAC, seguridad y piloto en terreno. * **Roles del equipo:**   + Product Owner (PO): define prioridades y valor de negocio.   + Scrum Master (SM): facilita ceremonias y elimina impedimentos.   + Team Lead (TL): coordina el equipo técnico.   + Desarrolladores (FE Web, FE Móvil, BE, DevOps, QA). * **Métodos de trabajo:** iteraciones cortas, revisión continua de requisitos, integración y despliegue continuo, retrospectivas y métricas ágiles (DORA). |

|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

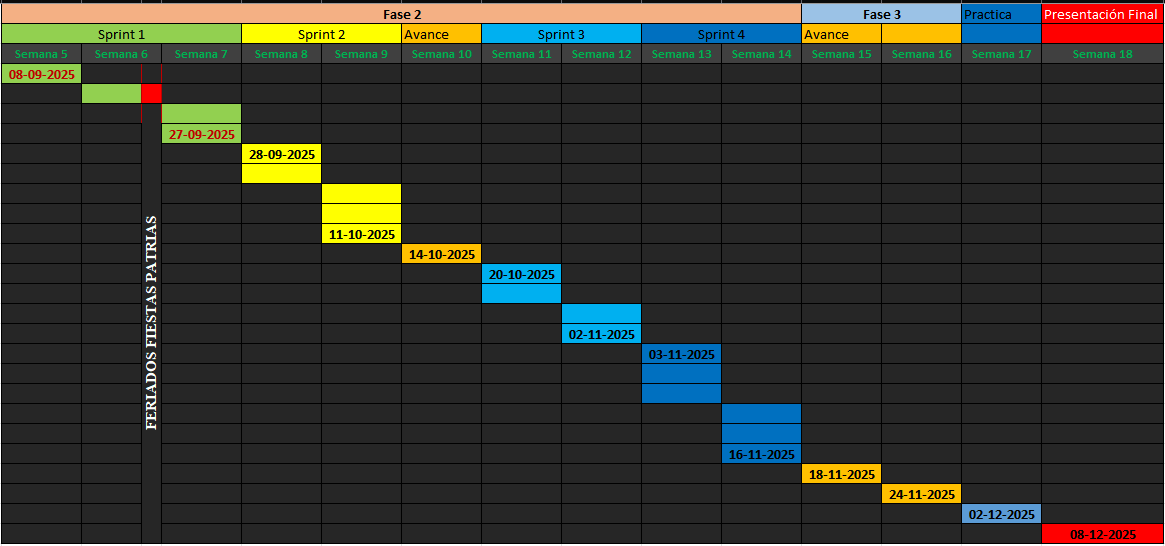
|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de evidencia  (avance o final) | Nombre de la evidencia | Descripción | Justificación |
| Avance  (Sprint Review) | Sprint Backlog actualizado | Lista de HU, RF cumplidos y entregables parciales | Evidencia de avance iterativo |
| Avance | Reportes de CI/CD | Resultados de pipelines con tests y despliegues | Garantizar calidad técnica y automatización |
| Final | RTM (Matriz de Trazabilidad) | Relación RF ↔ HU ↔ CP ↔ Módulo | Demuestra cumplimiento de requisitos |
| Final | Demo funcional y piloto | Sistema desplegado y validado con usuarios reales | Evidencia de la factibilidad y utilidad del sistema |

|  |
| --- |
| **7. Plan de Trabajo** |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia / Unidad | Nombre Actividad / Tarea | Descripción de la Actividad | Recursos | Duración | Responsable | Observaciones |
| Desarrollo de software, integración cloud | Registro de incidente en app móvil | Diseñar e implementar formulario con GPS, timestamp y modo offline-first | React, Capacitor, GPS API | 2 sem (Sprint 1) | FE Móvil | Riesgo: baja conectividad, mitigado con cola offline |
| Seguridad informática, bases de datos | Persistencia en API/DB | Implementar Endpoint POST /incidents y persistencia en MongoDB Atlas con folio único | Node.js, Express, MongoDB Atlas | 3 sem (Sprint 1) | BE, DevOps | Mitigar riesgo de pérdida de datos con Backup’s |
| Integración IA, automatización | Clasificación automática de incidentes | Configurar flujo n8n con OpenAI para asignar severidad y resumen automático | n8n, OpenAI API | 2 sem (Sprint 2) | DevOps, BE | Requiere tokens API; riesgo de costos |
| Desarrollo móvil / Backend | Evidencias en Storage | Configurar signed URL’s y subida segura de fotos/videos a Firebase Storage | Firebase, Cloud Storage | 2 sem (Sprint 2) | FE Móvil, BE | Riesgo de latencia en carga de archivos |
| Desarrollo Frontend, usabilidad | Dashboard básico | Implementar vista web con tabla de incidentes, filtros mínimos y búsqueda por folio | React, Tailwind, API REST | 2 sem (Sprint 2) | FE Web | Requiere datos de prueba en API |
| Automatización, comunicación digital | Notificaciones automáticas | Flujo de alertas críticas vía WhatsApp/Email con reintentos | n8n, WhatsApp Cloud API, SMTP | 3 sem (Sprint 3) | DevOps, BE | Riesgo: límites de mensajes API |
| Auditoría y cumplimiento | Logs y trazabilidad | Implementar registro de acciones con hash, timestamp y auditoría completa | Node.js, MongoDB, OTel | 3 sem (Sprint 3) | BE, QA | Verificar inmutabilidad en auditoría |
| Gestión documental, legal | Reporterías PDF oficiales | Generación de PDF con sello temporal y QR verificable | Node.js, jsPDF, Hash | 3 sem (Sprint 3) | BE, QA | Debe validarse legalmente con Stakeholders |
| Gestión de proyectos, KPIs | Dashboard avanzado con reportería | Implementar métricas y KPIs en tiempo real con filtros avanzados | React, Charts.js, API REST | 2 sem (Sprint 4) | FE Web | Depende de datos históricos cargados |
| Administración de sistemas | Gestión de usuarios y parámetros | Implementar RBAC con roles: Operador, Supervisor, Legal, Admin | Node.js, JWT/OAuth2, MongoDB | 3 sem (Sprint 4) | BE, DevOps | Revisión de políticas de acceso |
| Seguridad informática | Integraciones y fallback | Definir mecanismos de fallback en notificaciones y descargas de evidencias | n8n, APIs externas | 3 sem (Sprint 4) | DevOps | Validar resiliencia en caso de fallos |
| Gestión documental y normativas | Retención y archivado automático | Políticas de purga y almacenamiento según normativa legal | MongoDB TTL, n8n | 3 sem (Sprint 4) | BE, Admin | Revisión legal periódica |

|  |
| --- |
| **8. Carta Gantt** |
| Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título. |





**Anexos  
Informe Completo de Proyecto APT**



Informe Entrega Fase N°1   
**Definición Proyecto APT**

**Asignatura Capstone**

**Alumnos:**

José Almontes

Carlos Navarro

Giovanni Vargas

**Profesor:**

MAURICIO CARLOS FIGUEROA COLARTE

**Correo:**   
maur.figueroac@profesor.duoc.cl

**Asignatura:**

CAPSTONE\_007V  
2025\_2\_VM\_PTY4614\_24429585\_PCT

**Domingo 7 de septiembre de 2025**

**1. Índice**

|  |  |
| --- | --- |
| Descripción | Página |
| Portada | **1** |
| 1. Índice | **2** |
| 2. Introducción | **3** |
| 3. Abstract | **4** |
| 4. Contexto | **5 al 7** |
| 5. Problemática | **8 y 9** |
| 6. Necesidad | **10 y 11** |
| 7. Solución Propuesta | **12 al 16** |
| 8. Arquitectura de la Solución | **17 al 21** |
| 9. Metodología de Trabajo | **22 al 25** |
| 10. Roles y Responsabilidades | **26 al 28** |
| 11. Factibilidad | **29 al 33** |
| 12. Objetivos del Proyecto | **34 y 35** |
| 13. Requisitos | **36 al 39** |
| 14. Plan de Trabajo | **40 al 43** |
| 15. Evidencias | **44 al 49** |
| 16. Indicadores de Calidad | **50 al 53** |
| 17. Cronograma | **54 al 56** |
| 18. Conclusión General | **57** |
| 19. Reflexiones Individuales | **58 y 59** |
| 20. Bibliografía | **60** |
| 21. Anexos | **61 al 63** |

**2. Introducción**

El presente documento corresponde a la **Fase 1 del Capstone de Ingeniería en Informática**, donde se desarrolla el análisis preliminar, planificación y viabilidad del proyecto **EcoTrack**.

EcoTrack surge como una solución tecnológica frente a la creciente necesidad de **transparencia y trazabilidad en la gestión de incidentes**. Actualmente, los procesos manuales generan demoras, falta de evidencias y baja capacidad de auditoría. El proyecto propone digitalizar este flujo mediante una **aplicación móvil, un portal web y un Backend centralizado**, complementados con flujos de automatización (**n8n**) y herramientas de inteligencia artificial para clasificación automática de incidentes y notificaciones en tiempo real.

El documento está estructurado en diferentes apartados que abordan desde el **estudio de factibilidad técnica, económica, legal y operacional**, hasta el **plan de trabajo, gestión de interesados, riesgos y calidad**, siguiendo lineamientos de metodologías ágiles (Scrum) y marcos PMO.

Con este trabajo, se busca demostrar que el proyecto **es viable, estratégico y alineado tanto al perfil de egreso como a los intereses profesionales del equipo**, aportando valor a la industria de servicios y al ecosistema de gestión sustentable en Chile.



**3. Abstract**

EcoTrack es una plataforma digital diseñada para mejorar la trazabilidad, gestión y resolución de incidentes en la industria de gestión de residuos. El proyecto integra una **aplicación móvil (React + Capacitor)** que permite a los operadores registrar incidentes con geolocalización, sello de tiempo y evidencias multimedia (fotografías, documentos y PDF’s), junto con una **API Backend (Node.js + Express con MongoDB Atlas)** que garantiza el procesamiento centralizado, la auditoría de cambios y el almacenamiento seguro de las evidencias. Se incluye además un **Dashboard web (React)** para supervisores, que permite la supervisión de operaciones, el acceso a indicadores clave de desempeño (KPIs) y la validación del cumplimiento de acuerdos de nivel de servicio (SLA’s).

La solución incorpora **automatización mediante n8n**, conectándose con servicios externos como la **API de OpenAI** para la clasificación automática de severidad de incidentes y la generación de resúmenes, la **API de WhatsApp Cloud** para alertas críticas y **SMTP** para notificaciones formales vía correo electrónico. Esta arquitectura se despliega en **Firebase Hosting/Storage** con pipelines de integración y despliegue continuo (CI/CD) gestionados con GitHub Actions, asegurando entregas rápidas y confiables de los incrementos del producto.

EcoTrack responde a las limitaciones actuales de los **procesos manuales y fragmentados**, que aumentan los tiempos de respuesta y los riesgos legales, garantizando **notificaciones en tiempo real, custodia confiable de evidencias y flujos de trabajo automatizados**. Su diseño está alineado con los estándares de la industria en **ciberseguridad, computación en la nube, inteligencia artificial y gestión ágil de proyectos**, lo que hace que el proyecto sea técnica y profesionalmente viable, además de altamente relevante para el campo laboral de la informática.

**4. Contexto**

**4.1 Situación actual en la gestión de siniestros**

En la operación diaria de gestión de residuos, los **siniestros/incidentes** (derrames, desbordes, contenedores dañados, rutas interrumpidas, accidentes menores, hallazgos de riesgo) se reportan y atienden con **procesos mayormente manuales y fragmentados**. Esto genera brechas de trazabilidad, tiempos de respuesta altos y exposición a observaciones en auditorías.

**Cómo ocurre hoy (patrones comunes):**

* **Captura en terreno dispersa:** el Operador registra el incidente con llamadas telefónicas, mensajes de **WhatsApp** y fotografías almacenadas en el teléfono; la **geolocalización** no siempre se adjunta y el **sello temporal** no queda normalizado.
* **Registro manual en oficina:** el Supervisor transcribe a **planillas Excel** o formularios internos; el **folio** se asigna manualmente y no existe un repositorio único de evidencias (fotos, PDF’s).
* **Derivación ad-hoc:** avisos por correo/WhatsApp sin flujos definidos; no hay **priorización por severidad** ni confirmación automatizada de recepción.
* **Evidencias inconexas:** fotos, videos y documentos quedan **en múltiples ubicaciones** (dispositivos personales, nubes compartidas), dificultando la **custodia** y la **integridad** de la información.
* **Seguimiento y cierre manual:** el estado del incidente depende de comunicaciones interpersonales; se pierde el **hilo de auditoría** (quién hizo qué y cuándo).
* **Reportabilidad tardía:** consolidar información para **auditorías** o comités internos requiere tiempo, y aumenta el riesgo de **inconsistencias**.

**Dolores operativos y de cumplimiento:**

* **Tiempos de respuesta elevados**: asignación y resolución dependen de personas y disponibilidad; no hay **SLA** automatizados.
* **Riesgo de extravío o manipulación** de evidencias (sin control de versiones, sin hash ni logs de auditoría).
* **Duplicidad/retrabajo:** incidentes repetidos, información inconsistente entre equipos.
* **Falta de indicadores en línea:** **TTA/TTR, % de incidentes con evidencia completa, severidad por zona** no se visualizan en tiempo real.
* **Conectividad irregular en terreno:** zonas sin cobertura impiden informar en el momento; no existe un **modo offline** con sincronización confiable.
* **Heterogeneidad de dispositivos:** distintos teléfonos, versiones y permisos (GPS/cámara) afectan la **calidad de datos**.

**Requisitos regulatorios y contractuales relevantes:**

* **Trazabilidad completa** del ciclo (creación → atención → cierre) con evidencias verificables.
* **Custodia e integridad**: control de acceso por roles, **logs de auditoría** y **retención** por períodos definidos (p. ej., 3–5 años en incidentes críticos).
* **Transparencia y respuesta oportuna** frente a autoridades ambientales/municipales y clientes (licitaciones, inspecciones, auditorías).
* **Protección de datos**: controles de seguridad (cifrado en tránsito y en reposo, mínimos privilegios, política de permisos).

**Línea base recomendada (a validar en levantamiento):** TTA (creación→asignación), TTR (creación→resolución), **% de incidentes con evidencias completas** (foto + descripción + ubicación), **% de alertas entregadas a tiempo** y **tasa de reincidencia** por causa/zona.

**4.2 Relevancia en la industria (Cosemar SPA)**

La gestión de residuos opera en **entornos distribuidos**, con **múltiples cuadrillas**, alta exposición a incidentes y **exigencias de cumplimiento** crecientes (contratos públicos/privados, fiscalizaciones, estándares de sistemas de gestión). En este contexto:

**Para el sector (tendencias y exigencias):**

* **Digitalización y movilidad**: registrar en terreno con **geolocalización y sello temporal**, estandarizar datos y **automatizar** la notificación/derivación.
* **Evidencia probatoria**: **cadena de custodia**, integridad, y **retención** para responder a auditorías y reclamos.
* **Indicadores operacionales/KPI**: **TTR, TTA, severidad, reincidencia, cumplimiento de SLA** como base de mejora continua y competitividad.
* **Automatización e IA**: **clasificación automática** por severidad/criticidad, resúmenes y **enrutamiento inteligente** de incidentes.
* **Interoperabilidad**: integraciones con mensajería (WhatsApp/SMTP), nubes de almacenamiento, y tableros de control.

**Para Cosemar SPA (impacto específico):**

* **Reducción de tiempos** y **mejor priorización**: clasificación de severidad y enrutamiento inmediato reducen TTA/TTR y mejoran la atención en terreno.
* **Trazabilidad de punta a punta**: evidencias centralizadas, **PDF oficiales** con folio/sello de tiempo y **auditLogs** fortalecen la defensa frente a reclamos.
* **Cumplimiento y diferenciación**: **KPIs en tiempo real**, reportes formales y procesos auditables aumentan la **confianza del cliente** y la **competitividad en licitaciones**.
* **Eficiencia operativa**: menos retrabajo/duplicidad, mejor utilización de cuadrillas y mejor visibilidad para la jefatura.
* **Soporte a sistemas de gestión**: alineamiento con prácticas de **calidad, ambiente y seguridad** (p. ej., marcos ISO 9001/14001/45001 como referencia de buenas prácticas), facilitando auditorías.
* **Escalabilidad y resiliencia**: arquitectura cloud y **CI/CD** permiten crecer por zonas/contratos y recuperar servicio con rapidez (MTTR bajo).

**Riesgos de no modernizar:**

* **Mayor exposición a multas** y observaciones por falta de evidencia sólida.
* **Pérdida de contratos** por no cumplir SLA’s ni demostrar trazabilidad.
* **Sobrecarga de supervisores** y decisiones con información incompleta.
* **Deterioro reputacional** ante clientes y autoridades.

**Conclusión del contexto:**

El estado actual de la gestión de siniestros – Registro manual, fragmentado y con baja trazabilidad - **ya no es suficiente** para los niveles de exigencia del sector. Para Cosemar SPA, una solución como **EcoTrack** es **estratégica**: habilita **respuesta rápida**, **evidencia confiable**, **cumplimiento** y **mejora continua** con datos en tiempo real, alineándose con la transformación digital de la industria ambiental.

**5. Problemática**

La gestión de incidentes y operativos en empresas de servicios como **Cosemar SPA** enfrenta limitaciones significativas que impactan directamente en la **eficiencia operativa**, la **trazabilidad de la información** y el **cumplimiento normativo**.

**5.1 Falta de trazabilidad digital en incidentes**

Actualmente, los incidentes (siniestros menores, derrames, obstrucciones de ruta, fallas de contenedores, hallazgos de riesgo) son reportados y gestionados de manera manual, con **múltiples brechas en la cadena de registro y control**:

* **Captura en terreno no estandarizada:** los operadores dependen de **WhatsApp, llamadas telefónicas o notas manuscritas**, lo que provoca pérdida de datos críticos como la **ubicación GPS precisa**, la hora exacta del evento o la secuencia cronológica de las acciones realizadas.
* **Evidencias dispersas y sin custodia formal:** fotografías y documentos asociados a un incidente quedan en **dispositivos personales**, carpetas compartidas o correos aislados, dificultando demostrar la **integridad y autenticidad** de la evidencia frente a auditorías o reclamos.
* **Ausencia de folios y sellos temporales oficiales:** el proceso carece de un sistema que garantice la **unicidad del registro**, lo que genera duplicidades y complica la trazabilidad del ciclo completo del incidente (creación → atención → cierre).
* **Falta de indicadores en tiempo real:** métricas clave como el **TTR (tiempo de resolución)**, el **TTA (tiempo de asignación)** o el **porcentaje de incidentes con evidencias completas** no se generan automáticamente, imposibilitando el monitoreo proactivo y la mejora continua.

En consecuencia, **la organización carece de un repositorio centralizado y seguro**, lo que expone a la empresa a **riesgos legales, contractuales y reputacionales** por no poder demostrar la gestión adecuada de cada incidente.

**5.2 Procesos manuales y desconexión entre áreas (operaciones, legal, supervisión)**

La segunda dimensión crítica es la **fragmentación de los procesos** y la ausencia de integración entre las áreas involucradas:

* **Operaciones:** los equipos en terreno dependen de reportes manuales que llegan con retraso o incompletos. Esto limita la capacidad de **asignar rápidamente recursos** y de priorizar incidentes según criticidad.
* **Supervisión:** los supervisores reciben información por distintos canales (mensajería, llamadas, correos), lo que dificulta consolidar datos y tomar decisiones basadas en evidencias confiables.
* **Área legal:** en caso de reclamos, auditorías o litigios, la información no está estructurada ni auditada; se requiere un **proceso manual y tardío** para recopilar evidencias dispersas, aumentando la probabilidad de sanciones.
* **Falta de flujos automatizados:** no existen mecanismos automáticos para **notificar incidentes críticos**, **enviar alertas inmediatas** o **activar protocolos definidos** (por ejemplo, correo a legal o alerta vía WhatsApp al supervisor responsable).
* **Descoordinación interáreas:** la comunicación depende de la **buena voluntad y disponibilidad de las personas**, sin un sistema que garantice la trazabilidad de responsabilidades, el cumplimiento de SLA o la rendición de cuentas.

**5.3 Impacto de la problemática**

* **Operativo:** tiempos de respuesta altos, retrabajo y duplicidad de esfuerzos.
* **Legal:** exposición a sanciones por falta de evidencia robusta y custodiada.
* **Económico:** costos adicionales por ineficiencias, multas o pérdida de contratos.
* **Reputacional:** debilitamiento de la confianza de clientes y autoridades frente a incumplimientos o falta de transparencia.

**Conclusión de la problemática**

La **ausencia de una trazabilidad digital confiable** y la **fragmentación de procesos manuales entre operaciones, supervisión y legal** generan un entorno vulnerable, donde los incidentes no se gestionan con la rapidez, transparencia y respaldo documental necesarios. Esto afecta no solo la **eficiencia operativa** de Cosemar SPA, sino también su **capacidad de cumplir estándares legales y contractuales**, poniendo en riesgo la sostenibilidad y competitividad de la organización.

**6. Necesidad**

La problemática identificada en la gestión actual de siniestros e incidentes en Cosemar SPA revela **brechas críticas de trazabilidad, comunicación y control de evidencias**. Para superarlas, es necesario implementar una solución tecnológica integral que permita **digitalizar procesos, automatizar flujos de notificación y garantizar Reportabilidad Auditable**, con un impacto directo en la eficiencia operativa y el cumplimiento normativo.

**6.1 Digitalizar la captura de incidentes con evidencias**

Actualmente, los incidentes son reportados en terreno mediante **canales informales y no estandarizados** (WhatsApp, llamadas telefónicas, notas en papel), lo que dificulta asegurar integridad y completitud de los registros. Por ello, se requiere:

* Una **aplicación móvil multiplataforma (React + Capacitor)** que permita al operador **registrar incidentes en tiempo real**.
* **Campos obligatorios y estandarizados** (fecha/hora automática, geolocalización, tipo de incidente, severidad inicial, área responsable).
* **Carga de evidencias multimedia**: fotografías, videos, documentos adjuntos y formularios en PDF.
* **Modo offline** con sincronización diferida para zonas con baja conectividad, evitando pérdida de información crítica.
* **Asignación automática de folio y sello de tiempo**, garantizando unicidad y trazabilidad del registro desde su origen.

Con esta digitalización, se establece una **fuente única de verdad** que centraliza incidentes y asegura la **custodia formal de las evidencias**.

**6.2 Asegurar notificaciones inmediatas (WhatsApp / Email)**

Uno de los principales problemas es la demora en la comunicación de incidentes críticos. Para resolverlo, se plantea la integración de un **sistema automatizado de notificaciones inmediatas** mediante:

* **WhatsApp Cloud API:** envío de alertas críticas a supervisores y jefaturas en terreno, con detalles básicos del incidente (ubicación, tipo, foto adjunta).
* **SMTP/Email corporativo:** notificación formal a áreas clave (Legal, Prevención de Riesgos, Gerencia) con plantillas predefinidas que incluyen datos completos y evidencias asociadas.
* **Flujos de orquestación en n8n:** definición de reglas que disparan notificaciones según criterios de severidad, asegurando que ningún incidente crítico quede sin atender.
* **Confirmación de recepción:** mecanismos de acuse de recibo, que permiten validar que el mensaje fue entregado y leído.

Este componente garantiza una **reducción del Tiempo de Asignación (TTA)**, alineando la gestión a estándares de servicio y cumplimiento contractual.

**6.3 Proveer trazabilidad y reportes auditables**

La gestión actual carece de visibilidad consolidada y de evidencias confiables para auditorías. Para cerrar esta brecha, es necesario implementar:

* **Backend centralizado (Node.js + MongoDB Atlas):** registro único con auditoría de cambios (**auditLogs**) que almacene cada acción (creación, modificación, cierre) junto con usuario, fecha y hora.
* **Dashboard web (React):** acceso para supervisores con **KPIs en tiempo real** (TTR, TTA, % incidentes con evidencias completas, severidad por zona, tasa de reincidencia).
* **Generación de reportes PDF oficiales** con folio y sello temporal, válidos como evidencia en inspecciones y auditorías externas.
* **Políticas de retención y seguridad:** almacenamiento cifrado (TLS y at-rest), control de acceso por roles (RBAC: Operador, Supervisor, Legal, Admin) y conservación de evidencias por períodos normativos (3–5 años).
* **Integración con herramientas de observabilidad** (logs estructurados, métricas de negocio y analítica de uso) para asegurar trazabilidad y mejora continua.

Esto asegura la **transparencia, confiabilidad y defensa documental** de la organización ante clientes, autoridades y auditores externos.

**Conclusión de la necesidad**

Para que Cosemar SPA afronte los desafíos de eficiencia, cumplimiento y reputación, resulta imprescindible contar con una solución como **EcoTrack** que **digitalice la captura de incidentes**, **automatice notificaciones inmediatas** y **genere reportes auditables con trazabilidad completa**. Esta necesidad no es solo operativa, sino estratégica, ya que permite **fortalecer la confianza de clientes y reguladores**, mejorar la **capacidad de respuesta en terreno** y garantizar la **sostenibilidad competitiva** de la empresa en el sector ambiental.

**7. Solución Propuesta**

**7.1 Visión general**

**EcoTrack** es una plataforma de trazabilidad y gestión de incidentes compuesta por:

* **App móvil (React + Capacitor)** para registro en terreno con **offline-first**.
* **API Backend (Node.js + Express)** con **MongoDB Atlas** como base de datos transaccional.
* **n8n** para **automatización/orquestación** (IA, notificaciones, tareas programadas).
* **Dashboard web (React)** para supervisión, control y reportería.
* **Integraciones**: **OpenAI API** (clasificación), **WhatsApp Cloud API** (alertas), **SMTP** (correos).
* **Auditoría y respaldo probatorio**: **auditLogs** inmutables y **PDF** con folio y **sello temporal**.

**7.2 App móvil (React + Capacitor)**

**Objetivo:** digitalizar la captura en terreno garantizando **calidad de datos**, **evidencias** y **sincronización segura** aun sin conectividad.

**Funciones clave:**

* **Formulario guiado** con campos obligatorios: tipo de incidente, descripción, severidad inicial, cliente/contrato, fotos, **GPS** y **timestamp** automático.
* **Evidencias**: captura de fotos/video/documentos; compresión de imágenes, borrado de EXIF sensible; pre-hash (**SHA-256**) para detectar duplicados.
* **Offline-first**: persistencia local (SQLite/IndexedDB vía Capacitor) + **cola de sincronización** con reintentos exponenciales y resolución de conflictos “last-writer-wins” con **audit Trail**.
* **Geolocalización**: obtención de coordenadas con verificación de precisión; indicador de exactitud y validación de permisos de SO.
* **Seguridad**: autenticación vía **JWT/OAuth2** (login contra Backend/Firebase Auth), **TLS**; cifrado local de tokens/cola; bloqueo por PIN/biometría (opcional).
* **UX operacional**: estados “Borrador → Listo para enviar → Enviado/Con error”; indicador de conectividad; reenvío manual.

**Flujo básico (pseudosecuencia):**

1. Operador crea incidente ⇒ app genera **folio local preliminar** y **timestamp T0**.
2. App adjunta evidencias + GPS ⇒ guarda en cola local (offline).
3. Al detectar conectividad ⇒ **POST /incidents** con payload y pre-hash de evidencias ⇒ Backend devuelve **FOLIO definitivo**.
4. App queda a la espera de resultado de automatizaciones (clasificación IA/notificaciones); muestra estado.

**7.3 API Backend (Node.js + Express)**

**Objetivo:** exponer servicios REST, aplicar reglas de negocio, custodiar datos y evidencias, **auditar** cada acción y **disparar** orquestaciones.

**Endpoints principales (ejemplos):**

* POST /Auth/login → emisión de **JWT** (roles: Operador, Supervisor, Legal, Admin).
* POST /incidents → crear incidente (valida esquema; asigna **FOLIO** ECO-YYYYMMDD-######; genera **auditLog**).
* GET /incidents?status=&severity=&from=&to=&zone= → listado con **paginación y filtros**.
* GET /incidents/:id / PATCH /incidents/:id → consulta/actualización con **RBAC** y **auditLogs**.
* POST /incidents/:id/evidence → subir evidencias (recibe **signed URL**; guarda metadatos y hash).
* POST /incidents/:id/close → cierra incidente; dispara generación de **PDF**.
* GET /reports/incidents/:id/pdf → descarga del **PDF con sello**.
* GET /audit-logs?entity=incident&id=... → consulta de auditoría (solo Supervisor/Legal/Admin).

**Modelo de datos (resumen MongoDB):**

// incidents

{

"\_id": "...",

"folio": "ECO-20250901-000123",

"type": "Derrame",

"description": "Derrame en ruta sector norte",

"severity": "HIGH", // actualizado por IA o supervisor

"status": "OPEN|IN\_PROGRESS|CLOSED",

"location": { "type":"Point", "coordinates":[-70.6,-33.4] },

"reportedBy": { "userId":"...", "name":"..." },

"assignedTo": { "userId":"...", "name":"..." },

"evidence": [

{"id":"...", "kind":"photo", "url":"...signed", "hash":"...", "ts":"2025-09-01T10:21:00Z"}

],

"sla": { "ttaSec": 1800, "ttrSec": 7200 },

"timestamps": { "createdAt":"...", "assignedAt":"...", "closedAt":"..." }

}

**Índices propuestos:**

* location **2dsphere** (búsqueda por zona), status+createdAt, severity+createdAt, folio **único**.

**Calidad/seguridad:**

* Validación de entrada (**Zod/Joi**), **rate-limit** (por IP/usuario), **CORS** restringido, sanitización y control de tamaño de archivos.
* **Swagger/OpenAPI** publicado en /docs (contrato vivo).
* **Logs estructurados** (pino/winston) con **correlationId** por request.

**7.4 Base de datos y almacenamiento de evidencias**

* **MongoDB Atlas**: repositorio transaccional (incidents, users, roles, notifications, auditLogs).
* **Almacenamiento de evidencias**:
  + **Firebase Storage** (primario) con **signed URL’s** de lectura temporal y reglas por **RBAC**.
  + Alternativa/Backup: **MongoDB Atlas Cloud Storage** para archivos adjuntos pesados.
* **Retención**: políticas por severidad (p. ej., **3–5 años** para críticos); Job de n8n para **archivar/purgar** con acta de auditoría.

**7.5 n8n – Automatización e integraciones**

**Objetivo:** desacoplar y orquestar tareas automáticas con reintentos, trazabilidad y fallback.

**Flujo A – Clasificación y alertas (alta severidad):**

1. **Webhook (n8n)** recibe evento incident.created desde backend.
2. **OpenAI Node**: prompt con descripción + metadatos ⇒ respuesta {severity: LOW|MEDIUM|HIGH, summary:"..."}.
3. **PATCH /incidents/:id** (backend): actualiza severidad y guarda **análisis IA**.
4. **IF severity == HIGH** ⇒
   * **WhatsApp Cloud API**: envía alerta a Supervisor/Jefatura con **ubicación + miniatura**.
   * **SMTP**: correo formal a Legal/Prevención con detalle y links a evidencias.
5. **Log** y **reintentos**: si falla WA, intenta **SMTP**; si falla ambos, notifica canal de soporte.

**Flujo B – Cierre y PDF:**

1. incident.closed ⇒ **HTTP Node**: POST /reports/pdf (backend renderiza plantilla).
2. Guarda PDF en **Storage** y **adjunta** al incidente.
3. **SMTP**: envía a solicitante y Legal; **audita** resultado en auditLogs.

**Flujo C – Mantenimiento:**

* Tareas programadas: **recordatorios de SLA**, archivado/retención, métricas diarias, verificación de **entregabilidad** (WA/email).

**7.6 Dashboard web (React)**

**Módulos principales:**

* **Bandeja de incidentes**: filtros por severidad/estado/fecha/zona; búsqueda por folio; **mapa** con cluster de puntos.
* **Detalle y gestión**: reasignación, comentarios, historial y **auditLogs** visibles.
* **KPIs y reportería**: **TTR/TTA**, % evidencias completas, tasa de alertas entregadas, reincidencia por causa/ubicación; exportación **CSV/XLSX**.
* **Administración (RBAC)**: usuarios, roles (Operador/Supervisor/Legal/Admin), plantillas de notificación, parámetros de severidad/SLA.
* **Evidencias**: visor con **signed URLs** y marcas de agua (folio y fecha).

**Criterios de usabilidad:**

* Respuesta < **300 ms** en vistas principales (p95), paginación por servidor, accesibilidad básica (WCAG AA), estados de carga y error claros.

**7.7 Integraciones**

**WhatsApp Cloud API**

* Uso de **message templates** aprobadas (ej.: incident\_alert\_high) con variables: {{folio}} {{tipo}} {{ubicacion}} {{link}}.
* **Payload ejemplo** (resumen):

{

"messaging\_product": "whatsapp",

"to": "56XXXXXXXXX",

"type": "template",

"template": { "name": "incident\_alert\_high", "language": { "code": "es" },

"components": [{ "type": "body", "parameters": [

{"type":"text","text":"ECO-20250901-000123"},

{"type":"text","text":"Derrame"},

{"type":"text","text":"-33.45,-70.64"},

{"type":"text","text":"https://app.ecotrack/incident/ECO-20250901-000123"}

]}]

}

}

**SMTP (email corporativo)**

* Plantillas **HTML** con folio, detalles y vínculos a evidencias/PDF; soporte de **adjuntos**.
* Cabeceras con **messageId/correlationId** para auditoría.

**OpenAI API (Clasificación IA)**

* **Prompt base**: *“Clasifica la severidad (LOW/MEDIUM/HIGH) según impacto, seguridad y urgencia. Devuelve JSON con campos severity y summary. Texto: <descripción> + metadatos (tipo, fotos, zona)”*.
* **Controles de costo/privacidad**: truncado de entrada, **Caching** de resultados por folio, no enviar datos personales innecesarios.

**7.8 Auditoría y reportes con validez probatoria**

**Audit Logs**

* Registro inmutable por evento: {when, who, role, action, entity, entityId, from→to, ip, deviceId, correlationId, hash}.
* Opción de **hash encadenado** (tipo blockchain ligero) para detectar alteraciones.
* Acceso **solo** para Supervisor/Legal/Admin; exportación **CSV/PDF**.

**PDF con sello temporal**

* **Contenido**: folio, fechas (creación/asignación/cierre), tipo, severidad, descripción, ubicación (mapa mini), responsables, **evidencias embebidas** o vinculadas, bitácora (auditLogs resumidos), **QR** con URL del incidente.
* **Sello de tiempo**: issuedAt (UTC), hash del documento y **código de verificación**.
* **Firma digital/huella** (opcional): firmado por clave del sistema y registrado en auditLogs.

**7.9 Seguridad y cumplimiento (resumen)**

* **Autenticación**: JWT/OAuth2; **refresh tokens**; revocación por rol.
* **Autorización (RBAC)**: Operador (crear/editar propios), Supervisor (gestiona/cierra), Legal (acceso a evidencias y PDF), Admin (parametría/usuarios).
* **Protecciones**: TLS, CSP/CORS, **rate limiting**, sanitización, validación estricta, **signed URLs** para evidencias, cifrado at-rest (Atlas/Storage).
* **Retención**: políticas por severidad; **jobs** de purga/archivo; registro de cadena de custodia.

**7.10 Criterios de aceptación clave (extracto)**

* Crear incidente **offline** y sincronizarlo correctamente con **folio definitivo**.
* Clasificación IA almacenada y visible; **alertas HIGH** enviadas por **WA** y **email** con acuse.
* **PDF con sello** disponible al cierre y enlazado al incidente.
* **KPIs** (TTR/TTA/% evidencias) visibles en **dashboard** y exportables.
* Toda acción relevante genera **auditLog** consultable por rol autorizado.

**8. Arquitectura de la Solución**

**8.1 Vista general (capas y componentes)**

* **Frontend**
  + **App móvil**: React + Capacitor (Android/iOS), *offline-first*, geolocalización, captura de evidencias (fotos/PDF), cola de sincronización.
  + **Web dashboard**: React (SPA) para supervisores/Legal/Admin (bandeja de incidentes, KPIs, filtros, exportaciones, visor de evidencias).
* **Backend/API**
  + **Node.js + Express** (REST) con **Swagger/OpenAPI**, orquestación de negocio, RBAC, emisión/validación de JWT, *audit logs*, generación de PDF.
* **Datos y evidencias**
  + **MongoDB Atlas** (incidents, users/roles, auditLogs, notifications), índices por severidad/fecha/estado y 2dsphere para geoconsultas.
  + **Firebase Storage** (primario) / **MongoDB Atlas Cloud Storage** (alterno) para evidencias; acceso por **URL firmada** y reglas por rol.
* **Automatización**
  + **n8n** para flujos: clasificación IA (OpenAI), notificaciones (WhatsApp/SMTP), recordatorios SLA, archivado/retención.
* **Integraciones externas**
  + **OpenAI API** (clasificación de severidad y resúmenes), **WhatsApp Cloud API** (alertas críticas), **SMTP** (correo formal).
* **Infra y entrega**
  + **Firebase Hosting** (SPA React), contenedores para API/n8n, **GitHub Actions** (CI/CD), logging centralizado, métricas y trazas.

**Principios**: modularidad, *12-factor*, seguridad por defecto, *shift-left* en calidad, *feature flags* para activar integraciones en producción.

**8.2 Frontend**

**App móvil (React + Capacitor)**

* **Módulos**: Login (JWT/Firebase Auth), Registro de incidente, Adjuntos, GPS, Sincronización, Historial local.
* **Offline-first**: almacenamiento local (SQLite/IndexedDB), cola con reintentos exponenciales, resolución de conflictos (*last writer wins*) y huella de cambios para auditoría.
* **Captura**: compresión de imagen, eliminación de EXIF sensible, *pre-hash* (SHA-256) para detectar duplicados antes de subir.
* **Seguridad**: TLS, almacenamiento seguro de token, *biometric/PIN* opcional, *rate-limit* de acciones locales (anti-spam).

**Web dashboard (React)**

* **Vistas**: Bandeja (filtros por severidad/estado/fecha/zona), Mapa (cluster), Detalle (historial + auditLogs), KPIs (TTR, TTA, % evidencias, reincidencia), Administración (RBAC, plantillas).
* **Rendimiento**: SSR opcional o *code-splitting*, paginación por servidor, p95 < 300 ms, caché de lista con *stale-while-revalidate*.

**8.3 Backend API (Node.js + Express)**

* **Autenticación**: JWT/OAuth2; *refresh tokens*; *revocation list*.
* **Autorización**: RBAC (Operador, Supervisor, Legal, Admin) en *middlewares* por ruta.
* **Contratos**: Swagger/OpenAPI en /docs; *tests de contrato* en CI.
* **Endpoints clave (ejemplos)**
  + POST /auth/login → emite JWT
  + POST /incidents → crear incidente (folio ECO-YYYYMMDD-######)
  + GET /incidents → filtros: status,severity,from,to,zone
  + POST /incidents/:id/evidence → obtiene **signed URL** y registra metadatos/hash
  + PATCH /incidents/:id → cambios con *audit log*
  + POST /incidents/:id/close → cierre + trigger PDF
  + GET /reports/incidents/:id/pdf → descarga de PDF sellado
  + GET /audit-logs (rol: Supervisor/Legal/Admin)
* **Protecciones**: *input validation* (Zod/Joi), *rate-limit* por IP/usuario, CORS restringido, *size limits* para payloads, *idempotency keys* en creación.

**8.4 Modelo de datos (resumen)**

**Colección incidents**

{

"\_id": "…",

"folio": "ECO-20250901-000123",

"type": "Derrame",

"description": "…",

"severity": "LOW|MEDIUM|HIGH",

"status": "OPEN|IN\_PROGRESS|CLOSED",

"location": { "type":"Point", "coordinates":[-70.6,-33.4] },

"reportedBy": { "userId":"…", "name":"…" },

"assignedTo": { "userId":"…", "name":"…" },

"evidence": [

{"id":"…","kind":"photo","url":"…","hash":"SHA256:…","ts":"2025-09-01T10:21:00Z"}

],

"sla": { "ttaSec": 1800, "ttrSec": 7200 },

"timestamps": { "createdAt":"…", "assignedAt":"…", "closedAt":"…" }

}

**Colección auditLogs**

{

"when":"2025-09-01T10:22:11Z",

"who":{"userId":"…","role":"Supervisor"},

"action":"INCIDENT\_UPDATED",

"entity":"incident",

"entityId":"…",

"from": {"status":"OPEN"}, "to":{"status":"IN\_PROGRESS"},

"ip":"…", "correlationId":"…", "hash":"…"

}

**Índices**: folio (único), status+createdAt, severity+createdAt, location 2dsphere.

**8.5 Almacenamiento de evidencias**

* **Primario**: Firebase Storage con **reglas por rol** y acceso por **Signed URLs** (expiración corta).
* **Alterno/archivado**: MongoDB Atlas Cloud Storage.
* **Políticas**: retención 3–5 años para críticos; *jobs* de n8n para archivado/purga con acta en auditLogs.

**8.6 Automatización (n8n) e integraciones**

**Flujos n8n**

* **incident.created** → OpenAI (clasificación) → PATCH severidad → si HIGH ⇒ WhatsApp + Email.
* **incident.closed** → generar PDF → adjuntar a incidente → enviar a interesados (Legal/solicitante).
* **Programados**: recordatorios SLA, *healthchecks*, métricas diarias, verificación de entregabilidad.

**Integraciones**

* **OpenAI API**: *prompt* con descripción + metadatos, devuelve {severity, summary} (cache por folio).
* **WhatsApp Cloud API**: *message templates* aprobadas (folio/tipo/ubicación/enlace).
* **SMTP/Email**: plantillas HTML con adjuntos y *correlationId*.

**8.7 Seguridad y cumplimiento**

* **Transporte**: TLS 1.2+, HSTS.
* **Datos**: cifrado at-rest (Atlas/Storage), principios de mínimo privilegio.
* **Acceso**: RBAC estricto; *scopes* por API; *admin actions* auditadas.
* **Protecciones**: CSP, CORS, *rate-limit*, sanitización, validación estrita, *secrets* en *vault* (no en repos).
* **Privacidad**: no enviar PII innecesaria a OpenAI/WA; *redaction* de datos sensibles en logs.
* **Cumplimiento**: cadena de custodia (auditLogs), políticas de retención, trazabilidad de cambios.

**8.8 Observabilidad**

* **Logs estructurados** (pino/winston) con correlationId y niveles (info/warn/error).
* **Métricas**: rendimiento (p95), negocio (TTR/TTA, % evidencias, tasa de alertas), DORA (deploys, MTTR).
* **Trazas**: OpenTelemetry (opcional) entre App → API → n8n.
* **Dashboards**: panel de KPIs y salud de integraciones.

**8.9 Ambientes y despliegue**

* **Entornos**: *Dev* (sandbox), *Staging* (demo/QA), *Prod* (rollout controlado).
* **Hosting**:
  + React Web → **Firebase Hosting**
  + API y n8n → contenedores (servicio gestionado/K8s/VM) con *autoscaling*.
* **Feature flags**: activar WA/OpenAI gradualmente en Prod.
* **Rollback**: despliegues versionados, *blue/green* o *canary* cuando aplique.

**8.10 CI/CD (GitHub Actions)**

**Pipeline CI (en PR)**

1. **Checkout + Node setup**
2. **Lint** (ESLint) y *type-check*
3. **Unit & Integration tests** (Jest/Supertest) → **coverage ≥80%**
4. **SAST** (CodeQL/semgrep)
5. **Build** (API/SPA)
6. **Contract tests** (OpenAPI)

**Pipeline CD**

* **Staging (auto)** tras aprobar PR:
  1. Publicar imágenes/artefactos
  2. Despliegue a Staging
  3. **E2E** (Playwright) + **DAST** básico
* **Producción (manual/PO)**: *approval gate*, *migraciones*, *smoke tests*, *post-deploy checks* (SLAs).

**8.11 Flujos de secuencia (resumen)**

**A) Crear incidente (offline/online)**

1. App guarda borrador + GPS + evidencias → cola local.
2. Conectividad OK → POST /incidents → folio definitivo.
3. API emite evento → n8n clasifica (OpenAI) → si HIGH: WA + Email.
4. App y Dashboard ven estado actualizado.

**B) Cierre y reporte**

1. Supervisor cierra → POST /incidents/:id/close.
2. API genera **PDF** con sello y QR → guarda en Storage.
3. n8n envía correo formal con adjunto → *audit log*.

**8.12 Rendimiento y escalabilidad**

* **Objetivos**: p95 < 300 ms en endpoints críticos; ≥ 200 usuarios concurrentes; uptime ≥ 99,5%.
* **Escala**: *horizontal scaling* de API; índices y *capped collections* para logs; *CDN* para SPA; *cache* liviana en consultas de lista/filtros.

**8.13 Gestión de configuración y secretos**

* Variables de entorno por entorno (12-factor), *secrets* en *vault* del proveedor o GitHub Secrets, rotación periódica, *least-privilege* en claves de OpenAI/WA/SMTP.

**9. Metodología de Trabajo**

**9.1 Marco metodológico**

Se adopta **Scrum** como marco de trabajo ágil, complementado con **Product Backlog Building (PBB)** para **construir y mantener** un backlog claro, priorizado por valor y con trazabilidad desde la necesidad hasta la evidencia de prueba.

* **Duración de iteraciones**: Sprint’s cortos con entregas incrementales del producto potencialmente desplegable.
* **Principios**: inspección y adaptación continua, foco en valor temprano, colaboración con Stakeholders, calidad “**shift-left**” y **DevSecOps** integrado.

**9.2 Justificación (complejidad, cambios frecuentes, valor temprano)**

* **Complejidad técnica**: integración de móvil/web, Backend, base de datos, n8n, IA (OpenAI) y mensajería (WhatsApp/SMTP) requiere **aprendizaje iterativo** y **entrega por capas** (vertical slices).
* **Cambios frecuentes**: criterios de severidad, SLA y requisitos de auditoría pueden **evolucionar con la operación y el área legal**; Scrum permite **repriorizar cada sprint** con bajo costo de cambio.
* **Necesidad de valor temprano**: MVP con **registro + evidencias + clasificación + notificaciones** para validar en terreno (piloto), medir **TTA/TTR** y reducir riesgo de adopción.
* **Trazabilidad académica/profesional**: artefactos Scrum y PBB permiten **evidencias** (RTM, backlog, DoR/DoD, métricas) exigidas por la asignatura y por PMO.

**9.3 Product Backlog Building (PBB): cómo construimos el backlog**

**Objetivo**: traducir necesidades del negocio en **historias claras, priorizadas y trazables** hasta la prueba y la métrica.

**Pasos y entregables PBB**

1. **Personas y Reglas de negocio**
   * Personas: Operador, Supervisor, Legal, Admin.
   * Reglas: clasificación por severidad, SLA, retención de evidencias, cadena de custodia.
2. **Mapa de Historias (Story Mapping)**
   * Flujo TO-BE: **Registrar → Clasificar → Notificar → Gestionar → Auditar → Cerrar → Reportar**.
   * Se identifican **épicas** y **cortes verticales** de MVP.
3. **Épicas del Proyecto**

* **Épica 1: Registro de incidentes en terreno.**  
  Permitir que los operadores capturen incidentes desde la aplicación móvil, incluso en modo offline, incluyendo ubicación GPS, sello temporal y evidencias multimedia (fotografías, documentos).
* **Épica 2: Clasificación automática de incidentes con IA.**  
  Incorporar un flujo de inteligencia artificial que analice la información registrada y determine la severidad y el resumen del incidente, agilizando la priorización de la atención.
* **Épica 3: Notificaciones críticas en tiempo real.**  
  Implementar la orquestación de alertas mediante WhatsApp y correo electrónico (SMTP) para informar a supervisores y áreas clave cuando se detectan incidentes de alta severidad.
* **Épica 4: Auditoría y trazabilidad completa.**  
  Garantizar la integridad y custodia de la información mediante registros de auditoría (auditLogs), control de cambios, generación de PDF’s con sello temporal y códigos QR que validen la autenticidad.
* **Épica 5: Reporterías y KPIs operativos.**  
  Proveer un Dashboard con indicadores clave (TTA, TTR, porcentaje de evidencias completas, reincidencia por zona) y herramientas de reportería exportable (CSV/PDF), que respalden la toma de decisiones y el cumplimiento de SLA.
* **Épica 6: Seguridad y control de acceso basado en roles (RBAC).**  
  Implementar un modelo de seguridad robusto que diferencie permisos y responsabilidades entre Operador, Supervisor, Legal y Administrador, asegurando confidencialidad, integridad y disponibilidad de la información.

1. **Historias de Usuario**
   * Formato “Como [rol], quiero [capacidad], para [valor]”.
   * **Criterios de aceptación en Gherkin** (Given/When/Then) para reducir ambigüedad y habilitar pruebas automatizadas.
2. **Priorización y tamaño**
   * **MoSCoW** (Must/Should/Could/Won’t) + **WSJF** (valor negocio, criticidad de cumplimiento, riesgo reducido, esfuerzo).
   * Estimación con **Fibonacci** (1–13) y **capacidad** del equipo por sprint (velocidad).
3. **Definición de Capacidad y MVP**
   * Se fija una velocidad base (empírica) y se **ajusta** desde el sprint 2.
   * MVP = registro + evidencias + clasificación IA + notificaciones + Dashboard básico.

**9.4 Ceremonias y cadencia (por sprint)**

* **Sprint Planning (4 h)**
  + Define **Objetivo del Sprint**, alcance y riesgos.
  + Selección de historias según velocidad y dependencias.
  + Desglose a **tareas ≤ 8 h** con responsables.
* **Daily (15 min)**
  + Sincronización; bloqueos y plan de las próximas 24 h.
  + Actualización visual del **Sprint Board**.
* **Backlog Refinement (1–2 h/semana)**
  + Partir épicas, **pulir criterios Gherkin**, estimar y ordenar por valor.
  + Revisión de riesgos/seguridad (amenazas STRIDE si aplica).
* **Sprint Review (1,5 h)**
  + **Demo del incremento** a Operaciones/TI/Legal; decisiones de priorización.
  + Captura de **Feedback** para el Product Backlog.
* **Retrospective (1 h)**
  + Acciones concretas de mejora de proceso/calidad para el **siguiente sprint**.

**Sincronía con calidad/DevOps**: cada PR ejecuta **Lint, Tests, SAST** y despliega a **Staging**; la Review se hace sobre el incremento en Staging.

**9.5 Artefactos y definiciones de calidad**

* **Product Backlog**
  + Única fuente de verdad; historias con valor de negocio, criterios Gherkin, dependencias y riesgos.
* **Sprint Backlog**
  + Historias **comprometidas** → tareas técnicas (≤ 8 h) con **ID-HU | Tarea | Responsable | Estimación(h) | Estado | Bloqueos**.
* **Incremento**
  + Potencialmente desplegable; uso de **feature flags** para activar integraciones gradualmente.
* **DoR (Definition of Ready)**
  + Historia con **usuario/valor**, **criterios Gherkin**, **mock/contrato API**, datos de prueba, riesgos conocidos y estimación.
* **DoD (Definition of Done)**
  + Código fusionado + **tests unitarios ≥ 80%** y **de integración/E2E clave**; **SAST/DAST** sin hallazgos críticos; documentación mínima; **logging** básico; **Deploy a Staging**; **demo** funcional.

**Evidencias requeridas (por sprint)**

* Sprint Backlog actualizado, **reporte CI/CD**, **RTM** (Req ↔ HU ↔ CP ↔ Módulo ↔ Estado), **Acta de Retrospectiva**.

**9.6 Roles (Scrum Team + habilidades)**

* **PO (Product Owner)**
  + Visiona el producto, **prioriza** por valor, **acepta** historias, mantiene relación con Stakeholders.
* **SM (Scrum Master)**
  + Facilita el marco, **remueve impedimentos**, tutela la mejora continua.
* **TL (Tech Lead)**
  + Define **arquitectura y estándares**, revisa PR críticos, guía prácticas de seguridad/calidad.
* **FE Web (React)**
  + Dashboard, KPIs, filtros, exportaciones, integraciones con API.
* **FE Móvil (React + Capacitor)**
  + Registro **offline-first**, GPS, adjuntos, sincronización.
* **BE (Node.js + Express)**
  + API REST (Swagger), reglas de negocio, **RBAC**, **auditLogs**, performance.
* **QA (Quality Assurance)**
  + Plan de pruebas, **Unit/Integration/E2E**, gestión de defectos, automatización, criterios Gherkin.
* **DevOps**
  + **CI/CD (GitHub Actions)**, entornos (Dev/Staging/Prod), observabilidad y seguridad cloud.

**RBAC del sistema** (en el producto): Operador, Supervisor, Legal, Admin (alineado con historias y pruebas de seguridad).

**9.7 Estrategia de calidad y DevSecOps (shift-left)**

* **Pruebas**: unitarias (Jest/Vitest), integración (Supertest/Postman runner), **E2E (Playwright)** en Staging.
* **Seguridad**: **Lint + SAST** (CodeQL/semgrep) por PR; **DAST** por sprint; dependencia segura (npm audit/OWASP dep-check).
* **Contratos API**: Swagger/OpenAPI + **tests de contrato** en CI.
* **Performance**: objetivos p95 < **300 ms** en endpoints críticos (registro/clasificación/notificación).
* **Observabilidad**: logs estructurados, trazas (OTel), KPIs de negocio (**TTR/TTA/% evidencias**) y DORA (Deploy, MTTR, tasa de fallas).

**9.8 Métricas de proceso y gobernanza**

* **Proceso (Agile)**: Velocidad, **Burndown**, **Cumulative Flow**, lead/cycle time, defectos por sprint.
* **DevOps (DORA)**: frecuencia de despliegue (≥ 1/sprint), tiempo de cambio (≤ 1 día a Staging), **MTTR < 4 h**, tasa de fallas < 15%.
* **Producto**: **TTR/TTA** (↓ ≥ 60%), **% evidencias completas ≥ 95%**, tasa de entrega de alertas ≥ 95%, uptime ≥ 99,5%.
* **Gobernanza**: revisiones quincenales con Stakeholders; **Change Huddle** (PO+SM+TL) para solicitudes de cambio; **ADR** para decisiones arquitectónicas.

**10. Roles y Responsabilidades**

**10.1 Roles del equipo Scrum (desarrollo del proyecto)**

El proyecto **EcoTrack** se gestiona con un **equipo multidisciplinario** que cubre todas las áreas necesarias para el diseño, implementación, pruebas y despliegue de la solución. Cada rol tiene responsabilidades claras que garantizan la trazabilidad y la calidad del producto.

**Product Owner (PO)**

* Responsable de la **visión del producto** y del **alineamiento con las necesidades del negocio y de Cosemar SPA**.
* Define y prioriza el **Product Backlog** usando criterios de valor, riesgo y urgencia.
* Valida los incrementos entregados en cada sprint y **acepta historias de usuario**.
* Mantiene la comunicación directa con Stakeholders (Operaciones, Supervisión, Legal y Gerencia).

**Scrum Master (SM)**

* Facilita la aplicación de la metodología **Scrum**, asegurando la correcta ejecución de ceremonias.
* Elimina impedimentos que afecten la productividad del equipo.
* Promueve la **mejora continua** en procesos, calidad y colaboración.
* Se asegura de que las definiciones de **DoR (Definition of Ready)** y **DoD (Definition of Done)** sean respetadas.

**Tech Lead (TL)**

* Define la **arquitectura técnica** del sistema y establece **estándares de calidad, seguridad y performance**.
* Supervisa revisiones de código críticas (Pull Request’s).
* Asegura la correcta integración de los distintos módulos (App móvil, API Backend, Dashboard web, n8n).
* Apoya en la estimación técnica y en la resolución de problemas complejos.

**Frontend Web Developer (FE Web)**

* Desarrolla el **Dashboard React** para supervisores, legales y administradores.
* Implementa funcionalidades de gestión, visualización de KPIs y reportería.
* Se asegura de la **usabilidad** y **accesibilidad** en las interfaces de usuario.

**Frontend Mobile Developer (FE Móvil)**

* Construye la **aplicación móvil (React + Capacitor)** usada por los operadores en terreno.
* Implementa la captura de incidentes con **GPS, timestamp y evidencias multimedia**.
* Diseña y mantiene el **modo offline** y los procesos de sincronización.

**Backend Developer (BE)**

* Implementa la **API REST con Node.js + Express** y la base de datos en **MongoDB Atlas**.
* Define y ejecuta reglas de negocio, manejo de evidencias y generación de folios.
* Garantiza la **seguridad** (JWT/OAuth2, RBAC) y la **integridad de datos** (auditLogs).

**Quality Assurance (QA)**

* Diseña y ejecuta **pruebas unitarias, de integración y E2E**.
* Documenta casos de prueba y mantiene actualizada la **Matriz de Trazabilidad (RTM)**.
* Asegura que cada historia cumpla con criterios Gherkin antes de ser considerada “Done”.
* Registra defectos, da seguimiento y valida correcciones.

**DevOps Engineer**

* Diseña y mantiene la **pipeline CI/CD en GitHub Actions**.
* Administra entornos (Dev, Staging, Producción).
* Configura monitoreo, alertas y observabilidad.
* Asegura la **seguridad en despliegues cloud** y administra secretos de integraciones (WA, SMTP, OpenAI).

**10.2 Roles RBAC del sistema EcoTrack (usuarios finales)**

Además del equipo Scrum, el sistema EcoTrack implementa un modelo de **Roles-Based Access Control (RBAC)** para garantizar seguridad y trazabilidad en el uso de la plataforma. Cada usuario interactúa con permisos según su rol organizacional.

**Operador**

* **Crear y editar incidentes propios** desde la app móvil.
* Adjuntar evidencias multimedia (fotos, videos, PDF) con ubicación y timestamp.
* Sincronizar registros en modo offline.
* Consultar estado de sus incidentes.
* No tiene acceso a reportes globales ni a incidentes de otros operadores.

**Supervisor**

* **Gestionar incidentes creados por los operadores de su área**.
* Aprobar o rechazar cierres de incidentes.
* Reasignar responsables y dar seguimiento a incidentes críticos.
* Acceder a KPIs y al Dashboard de supervisión (TTR, TTA, % evidencias).
* Acceder a evidencias y auditLogs de incidentes de su zona.

**Legal**

* Consultar incidentes cerrados y activos relevantes a auditorías.
* Acceder a **todas las evidencias asociadas** (fotos, PDF, logs).
* Generar **reportes oficiales en PDF con sello temporal** para respaldar procesos legales o regulatorios.
* Validar cumplimiento de **políticas de retención y custodia**.

**Administrador (Admin)**

* Configurar **usuarios, roles y permisos** en el sistema.
* Administrar parámetros globales: SLA, plantillas de notificación, reglas de severidad.
* Monitorear la operación completa del sistema (incidentes, logs, integraciones).
* Acceder a todos los módulos (Operador, Supervisor, Legal) con permisos avanzados.

**10.3 Relación entre roles Scrum y roles RBAC**

* **PO y SM** trabajan con Stakeholders que corresponden a roles **Supervisor, Legal y Admin** para refinar el backlog.
* **FE Móvil y BE** implementan flujos críticos de **Operador**.
* **FE Web, QA y TL** se enfocan en funcionalidades de **Supervisor, Legal y Admin**.
* **DevOps** asegura que todos los roles del sistema operen de forma segura y con alta disponibilidad.

**11. Factibilidad**

**11.1 Factibilidad técnica**

**Conclusión:** **Viable**. El Stack propuesto es maduro, ampliamente adoptado y ya **probado en producción** en contextos similares. La arquitectura modular y desacoplada reduce el riesgo técnico y facilita la evolución por capas.

**Stack y pilares técnicos**

* **Frontend móvil (React + Capacitor)**: una sola base de código para Android/iOS, con **offline-first** (persistencia local, cola de sincronización, reintentos exponenciales) y acceso nativo a GPS/cámara.
* **Frontend web (React)**: Dashboard SPA con filtros, mapa, KPIs y exportaciones; paginación por servidor; **p95 < 300 ms** en vistas críticas como objetivo de rendimiento.
* **Backend (Node.js + Express)**: API REST con **Swagger/OpenAPI**, validación de entrada (Joi/Zod), **RBAC**, **auditLogs** y generación de **PDF** con folio y sello temporal.
* **Datos (MongoDB Atlas)**: colecciones incidents, auditLogs, users/roles; índices por status+createdAt, severity+createdAt y **2dsphere** para consultas geoespaciales.
* **Evidencias (Firebase Storage / Atlas Cloud Storage)**: acceso mediante **signed URL’s** con expiración corta y reglas por rol.
* **Automatización (n8n)**: **Webhook** desde la API; flujos para clasificación IA (OpenAI), notificaciones (WhatsApp/SMTP), recordatorios de SLA, archivado/retención.
* **Integraciones**:
  + **OpenAI API**: clasificación de severidad y resúmenes bajo **Prompt’s acotados** y **Caching** por folio.
  + **WhatsApp Cloud API**: plantillas aprobadas para alertas críticas.
  + **SMTP**: notificaciones formales y envío de PDF.
* **Calidad/seguridad integradas**: CI (lint, Unit/Integration, **Coverage ≥80%**), SAST (CodeQL/semgrep), DAST básico por sprint; TLS 1.2+, cifrado at-rest, **rate-limit**, CORS, CSP y gestión de secretos en vault/Secrets.

**Capacidad y desempeño esperados (supuestos de dimensionamiento)**

* 200–300 usuarios móviles ocasionales; 20–40 usuarios web concurrentes.
* 1.000–3.000 incidentes/mes; 2–5 evidencias promedio por incidente.
* **SLO**: uptime ≥ **99,5%**; TTR/TTA **↓ ≥ 60%** respecto la línea base; alertas críticas entregadas ≥ **95%** en ≤ **2 min**.
* Escalado horizontal de API; almacenamiento de evidencias elástico; colas de reintentos para picos.

**11.2 Factibilidad económica**

**Conclusión:** **Viable (costos contenidos)**. Se aprovechan **servicios gestionados** para reducir CAPEX y esfuerzo de operación.

**Costo operativo mensual estimado (OPEX)**

* **MongoDB Atlas**: **$200.000 CLP** (plan inicial + Backup’s).
* **Firebase Hosting/Storage**: **$100.000 CLP** (hosting SPA + evidencias de media frecuencia).
* **OpenAI API**: **$150.000 CLP** (Prompt’s optimizados, truncado de contexto y **Caching** por folio).
* **WhatsApp Cloud API + SMTP**: **$100.000 CLP** (mensajería crítica y correo formal).
* **n8n + otros**: **$100.000 CLP** (orquestación y tareas programadas).  
  **Total estimado:** **≈ $650.000 CLP/mes**.

*Notas:* valores referenciales para entorno productivo de baja/mediana escala. En Capstone y piloto se pueden usar **tiers gratuitos** o menores, reduciendo el gasto.

**Costos no recurrentes (bajos)**

* Plantillas de correo y WhatsApp, capacitación breve de usuarios (Onboarding) y documentación operativa.

**11.3 Factibilidad legal y de cumplimiento**

**Conclusión:** **Viable** si se aplican los controles definidos.

**Controles clave**

* **Consentimiento informado** y **propósito explícito** en el registro de incidentes.
* **Retención** de evidencias por severidad (p. ej., **3–5 años** para críticos); archivado/purga automatizados con **acta en auditLogs**.
* **Seguridad**: cifrado **TLS** en tránsito y **at-rest** en Atlas/Storage; **RBAC** (Operador, Supervisor, Legal, Admin); **principio de mínimo privilegio**; **signed URL’s** para evidencias.
* **Auditabilidad**: **auditLogs** inmutables (quién, qué, cuándo, desde dónde), PDF’s con **folio, sello temporal y hash**; trazabilidad de cambios.
* **Privacidad**: evitar envío de PII innecesaria a OpenAI/WA; **redacción** de datos sensibles en logs; gestión de **derechos ARCO** si aplica.

**11.4 Factibilidad de recursos (equipo)**

**Conclusión:** **Suficiente** para MVP y piloto.

**Equipo Scrum (8 roles)**

* **PO, SM, TL, FE Web, FE Móvil, BE, QA, DevOps**.
* Cobertura completa de disciplinas: arquitectura, desarrollo móvil/web, API, QA automatizado y **CI/CD**.
* **Velocidad inicial** estimada y ajuste empírico desde Sprint 2.

**11.5 Factibilidad temporal**

**Conclusión:** **Alcanzable** en el marco de la asignatura.

**Sprint 1 – Base del MVP (Semanas 4 a 6)**

**Duración:** 3 semanas 3 días

**Actividades:**

* App móvil con **registro de incidentes**, geolocalización (GPS) y sincronización offline.
* API y base de datos inicial (Node.js + MongoDB).
* Pruebas unitarias iniciales.

**Entregable:**

Base del **MVP funcional** con registro de incidentes y Backend operativo.

**Sprint 2 – MVP Completo (Semanas 7 a 8)**

**Duración:** 2 semanas

**Actividades:**

* Evidencias en **Firebase Storage**.
* Clasificación de incidentes con **IA (OpenAI)**.
* Dashboard básico con métricas iniciales.
* Pruebas de integración.

**Entregable:**

**MVP completo** con evidencias, clasificación inteligente y Dashboard inicial.

**Sprint 3 – Mejoras Avanzadas (Semanas 9 a 10)**

**Duración:** 2 semanas

**Actividades:**

* Notificaciones automáticas vía **WhatsApp/SMTP**.
* Reportes **PDF con folio y sello temporal**.
* KPIs en Dashboard (TTR, TTA, % evidencias).
* Filtros avanzados en panel.
* Auditoría de logs y pruebas E2E iniciales.

**Entregable:**

Plataforma mejorada con **reportes, KPIs y notificaciones**.

**Sprint 4 – Seguridad y Piloto Final (Semana 11)**

**Duración:** 1 semana

**Actividades:**

* **RBAC completo** (roles y permisos).
* Hardening de seguridad y performance (**p95 < 300ms**).
* Monitoreo y logging avanzado.
* **Piloto en terreno** con usuarios reales.
* Ajustes finales y entrega del proyecto.

**Entregable:**

**Versión final lista para producción**, validada en piloto con seguridad y rendimiento optimizados.

**Entregables por sprint (evidencias)**

* **Sprint Backlog**, **demo del incremento**, **reporte CI/CD**, **RTM actualizado**, **acta de retrospectiva**.

**11.6 Riesgos y mitigación**

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgo | Impacto | Prob. | Mitigación |
| Conectividad en terreno (zonas sin señal) | Pérdida/demora de registro | Media | **Offline-first**, cola de sincronización, reintentos exponenciales, validación de GPS y almacenamiento local cifrado. |
| Costos de APIs externas (OpenAI/WA) | Sobrepasar presupuesto | Media | **Capping** por entorno, **Caching** por folio, Prompt’s truncados, *backoff*; **fallback** a email si falla WA; revisión mensual de consumo. |
| Resistencia al cambio (adopción) | Baja utilización | Media | **Capacitación** breve, manuales con capturas, soporte de **primer mes**, Feedback en Review; líderes de cuadrilla como champions. |
| Seguridad/privacidad | Incidente de datos | Baja | TLS, cifrado at-rest, RBAC estricto, secretos en vault, **SAST/DAST**, logging con **redacción** de PII, pruebas de autorización. |
| Calidad de datos (fotos/EXIF/ubicación) | Evidencias inválidas | Media | Compresión y limpieza de EXIF, campos obligatorios, validación de precisión GPS, Checklist en la app. |
| Escalabilidad/Performance | Degradación bajo carga | Baja | Índices en Atlas, paginación server-side, cache liviana, escalado horizontal de API, objetivos **p95 < 300 ms**. |
| Disponibilidad de integraciones | Falla de terceros | Media | **Circuit breaker**, reintentos, colas en n8n, fallback (solo email), alertas de salud y *runbooks*. |

**11.7 Criterios de éxito de la factibilidad (medición)**

* **Disponibilidad** ≥ 99,5% (SLA mensual).
* **TTA/TTR**: reducción **≥ 60%** vs. línea base al finalizar Sprint 5 (piloto).
* **% de incidentes con evidencia completa** ≥ **95%**.
* **Entrega de alertas críticas** ≥ **95%** ≤ **2 min**.
* **DORA**: ≥ 1 despliegue por sprint; **MTTR < 4 h**; tasa de fallas < 15%.

**12. Objetivos del Proyecto (SMART)**

**12.1 Objetivo general**

Desarrollar una plataforma digital integral de trazabilidad de incidentes (EcoTrack), que permita registrar, clasificar, notificar, auditar y reportar incidentes de manera segura, eficiente y en tiempo real, mejorando en al menos un 60% los tiempos de asignación (TTA) y resolución (TTR) en comparación con los procesos manuales actuales, dentro de un plazo de 11 semanas.

* **Específico:** se centra en la creación de la plataforma digital (móvil + web + Backend + automatización).
* **Medible:** éxito validado con métricas de TTA y TTR, % de evidencias completas y uptime ≥ 99,5%.
* **Alcanzable:** recursos humanos (8 roles Scrum) y Stack probado (React, Node.js, MongoDB, Firebase, n8n).
* **Relevante:** responde a una necesidad crítica en Cosemar SPA y el sector ambiental, alineado al perfil de egreso.
* **Temporal:** entregable en 11 semanas, dividido en 5 Sprint’s con hitos medibles.

**12.2 Objetivos específicos**

**OE1 – Registro digital de incidentes con geolocalización y evidencias**

* **Specific:** Implementar una aplicación móvil (React + Capacitor) que permita a los operadores registrar incidentes con GPS automático, timestamp y evidencias multimedia (fotos/PDF).
* **Measurable:** ≥ 95% de incidentes registrados deben contar con al menos una evidencia válida.
* **Achievable:** mediante el uso de almacenamiento local y sincronización offline/online.
* **Relevant:** asegura integridad de la captura y elimina pérdida de información.
* **Time-bound:** desplegado al finalizar el **Sprint 1**.

**OE2 – Clasificación automática de incidentes mediante IA**

* **Specific:** Integrar la API de OpenAI para clasificar incidentes en severidades (LOW, MEDIUM, HIGH) y generar resúmenes automáticos.
* **Measurable:** precisión aceptada ≥ 85% en pruebas comparativas con clasificación manual.
* **Achievable:** usando Prompt’s optimizados y Caching por folio en n8n.
* **Relevant:** reduce carga de supervisores y estandariza criterios.
* **Time-bound:** operativo al cierre del **Sprint 2 (MVP)**.

**OE3 – Notificaciones automáticas en tiempo real (WhatsApp/Email)**

* **Specific:** Configurar n8n para enviar notificaciones inmediatas vía WhatsApp Cloud API y SMTP cuando un incidente clasificado como crítico sea registrado.
* **Measurable:** ≥ 95% de alertas entregadas en ≤ 2 min.
* **Achievable:** mediante plantillas aprobadas y flujos con reintentos en n8n.
* **Relevant:** mejora el **Tiempo de Asignación (TTA)** y la reacción en campo.
* **Time-bound:** entregable en el **Sprint 3**.

**OE4 – Generación de reportes PDF con sello temporal**

* **Specific:** Implementar en el Backend un servicio para emitir reportes PDF con folio, sello temporal, evidencias y QR de validación.
* **Measurable:** 100% de incidentes cerrados deben generar un PDF válido y auditable.
* **Achievable:** usando librerías de PDF en Node.js y almacenamiento en Firebase Storage.
* **Relevant:** cumple exigencias legales y de auditoría.
* **Time-bound:** funcionalidad entregada en el **Sprint 4**.

**OE5 – Dashboard con KPIs y trazabilidad auditada**

* **Specific:** Construir un Dashboard web (React) para supervisores con indicadores clave (TTR, TTA, % de evidencias completas, reincidencias), filtros avanzados y exportaciones.
* **Measurable:** KPIs visibles en tiempo real; uptime ≥ 99,5%.
* **Achievable:** a través de APIs optimizadas, índices en MongoDB y visualización en React.
* **Relevant:** permite seguimiento de gestión y mejora continua.
* **Time-bound:** desplegado en el **Sprint 5 (piloto)**.

**13. Requisitos**

**13.1 Alcance y supuestos**

* **Ámbito:** Plataforma EcoTrack (App Móvil React+Capacitor, Web Dashboard React, API Node.js/Express, MongoDB Atlas, n8n, Integraciones OpenAI/WhatsApp/SMTP).
* **Usuarios (RBAC):** Operador, Supervisor, Legal, Admin.
* **Supuestos clave:** conectividad variable en terreno (se exige **offline-first**), lineamientos de seguridad y retención definidos por la organización, uso de servicios gestionados (Firebase Storage/Mongo Atlas).

**13.2 Requisitos Funcionales (RF)**

**RF-01 — Registro de incidente (App móvil)**

* **Descripción:** El Operador puede crear un incidente con **folio automático**, fecha/hora, **GPS**, tipo, descripción, severidad tentativa y adjuntar evidencias.
* **Datos mínimos:** {tipo, descripción, ubicación(GPS), evidencias(≥1), severidad\_inicial, contrato/zona}
* **Criterios de aceptación (Gherkin):**

*Given* que estoy autenticado, *When* completo los campos obligatorios y adjunto ≥1 foto, *Then* el sistema guarda el incidente con **folio único** y **timestamp**.

**RF-02 — Adjuntar evidencias**

* **Descripción:** Fotos, videos y PDF se asocian al incidente; se calcula **hash** y se guardan metadatos (tamaño, tipo, fecha).
* **Reglas:** ≥1 evidencia obligatoria al crear; no se permite >25 MB por archivo (configurable).

**RF-03 — Geolocalización**

* **Descripción:** Captura coordenadas y precisión; si el usuario niega permisos, la app solicita nuevamente o permite ingreso manual (marcado como *no verificado*).

**RF-04 — Trabajo offline y sincronización**

* **Descripción:** La app funciona sin red; guarda en almacenamiento local y sincroniza automáticamente con **reintentos exponenciales**.
* **Estados:** Borrador → Listo para enviar → Enviado/Con error.

**RF-05 — Edición y seguimiento**

* **Descripción:** El Operador puede **editar** el incidente hasta que pase a *IN\_PROGRESS*; luego solo Supervisor/Legal.

**RF-06 — Clasificación automática (IA)**

* **Descripción:** Al crear un incidente, n8n invoca **OpenAI** con la descripción/metadatos; la API actualiza severity con **LOW/MEDIUM/HIGH** y un **resumen IA**.
* **Criterios:** El resultado IA queda **visible y auditable**; Supervisor puede **reclasificar**.

**RF-07 — Notificaciones automáticas**

* **Descripción:** Incidentes **HIGH** disparan **WhatsApp** (supervisores/jefatura) y **Email SMTP** (Legal/Prevención).
* **SLA notificación:** ≥95% entregadas en **≤2 min** (medido por logs de entrega).

**RF-08 — Gestión de incidentes (Supervisor)**

* **Descripción:** Listado, filtros (estado, severidad, fecha, zona), **reasignación**, comentarios, cambio de estado y aprobación de cierre.

**RF-09 — Auditoría (auditLogs)**

* **Descripción:** Toda acción relevante genera un registro: quién, rol, cuándo, acción, entidad, de→a, ip, correlationId, hash.

**RF-10 — Reporte PDF con sello**

* **Descripción:** Al cerrar, se genera **PDF oficial** con folio, datos, evidencias, bitácora resumida, **sello temporal** y **QR** de verificación.
* **Acceso:** descarga desde Web (Supervisor/Legal) y enlace en correo.

**RF-11 — Dashboard y KPIs (Web)**

* **Descripción:** Panel con **TTR, TTA, % evidencias completas, severidad por zona, reincidencia**.
* **Acciones:** filtros, ordenación, exportación **CSV/XLSX**.

**RF-12 — Búsqueda y filtros avanzados**

* **Descripción:** Por folio, rango de fechas, severidad, estado, zona/cliente; paginación y orden por fecha/severidad.

**RF-13 — Administración (RBAC)**

* **Descripción:** Alta/baja/edición de usuarios, asignación de roles, **parámetros** (SLA, severidades, plantillas de notificación).

**RF-14 — Versionado de API**

* **Descripción:** Endpoints versionados (/api/v1) y **Swagger/OpenAPI** público interno en /docs.

**RF-15 — Integraciones y fallbacks**

* **Descripción:** Si **WhatsApp** falla, se envía **Email**; si ambos fallan, se notifica canal de soporte y queda registro en **auditLogs**.

**RF-16 — Exportaciones y evidencias**

* **Descripción:** Exportar listados (CSV/XLSX) con filtros activos; descargar evidencias con **signed URL’s** (expiran).

**RF-17 — Políticas de retención**

* **Descripción:** Jobs de n8n para archivar/purgar tras 3–5 años según severidad; acta en **auditLogs**.

**13.3 Requisitos No Funcionales (RNF)**

**Disponibilidad y rendimiento**

* **RNF-01 — Disponibilidad:** **≥ 98%** mensual (SLA del servicio).
* **RNF-02 — Rendimiento API:** **p95 < 300 ms** en endpoints críticos (crear/listar incidentes, clasificar, notificar).
* **RNF-03 — Rendimiento Web:** tiempo de primera interacción aceptable en redes móviles (objetivo < 3 s en 4G).
* **RNF-04 — Concurrencia:** soportar ≥ 200 usuarios móviles/día y ≥ 40 usuarios web concurrentes (capacidad inicial).

**Seguridad y privacidad**

* **RNF-05 — Transporte y datos:** **TLS 1.2+**, cifrado **at-rest** (Atlas/Storage), **signed URL’s** para evidencias.
* **RNF-06 — Control de acceso:** **JWT/OAuth2** + **RBAC**; principio de **mínimo privilegio**; bloqueo de cuenta por intentos fallidos.
* **RNF-07 — Buenas prácticas OWASP:** validación estricta, sanitización, **CORS** restringido, **rate-limiting**, protección CSRF donde aplique.
* **RNF-08 — Privacidad:** no enviar PII innecesaria a OpenAI/WA; **redacción** de datos sensibles en logs; cumplimiento de solicitudes de acceso/eliminación cuando apliquen.

**Usabilidad y accesibilidad**

* **RNF-09 — Usabilidad móvil:** formularios guiados, mínimos toques, estados claros (offline/enviado/error), soporte **dark mode** (opcional).
* **RNF-10 — Accesibilidad:** cumplir **WCAG 2.1 AA** en la Web (contraste, foco, navegación por teclado, alternativas textuales).
* **RNF-11 — Multi-idioma (opcional):** estructura preparada para i18n/es-CL.

**Confiabilidad y observabilidad**

* **RNF-12 — Logs y trazas:** **logs estructurados** con correlationId; almacenamiento centralizado; retención acorde a políticas; **trazas OTel** opcionales.
* **RNF-13 — Métricas:** negocio (**TTR, TTA, % evidencias completas, tasa de alertas**), técnicas (latencia p95, errores 5xx), **DORA** (Deploy freq, MTTR, change fail rate).
* **RNF-14 — Backup’s:** políticas de respaldo de Atlas (automático) y verificación mensual de restauración.

**Escalabilidad, mantenibilidad y entrega**

* **RNF-15 — Escalabilidad:** **horizontal** para API; particionado lógico por zonas si crece el volumen; índices adecuados en Atlas (2dsphere, status + fecha, folio único).
* **RNF-16 — Mantenibilidad:** código con **linting**, estándares y **Coverage ≥ 80%** (API); documentación mínima por módulo; ADR para decisiones.
* **RNF-17 — CI/CD y *quality gates*:** Lint + Unit/Integration + **SAST** en cada PR; E2E (Staging) y **DAST** por sprint; no se permite merge si falla algún *gate*.
* **RNF-18 — Versionado y compatibilidad:** API **/v1** estable; cambios *breaking* pasan por *feature flag* y aviso previo.
* **RNF-19 — Portabilidad:** infraestructura como código (plantillas de despliegue reproducibles); variables por entorno.

**Interoperabilidad y entrega de notificaciones**

* **RNF-20 — Entregabilidad:** ≥ **95%** de alertas críticas se entregan en ≤ **2 min** (WA/Email); monitor de éxito/falla con reintentos.
* **RNF-21 — Idempotencia:** creación de incidentes soporta **idempotency keys** para evitar duplicados en reconexión.

**Integridad de evidencias**

* **RNF-22 — Hash e integridad:** cada evidencia almacena **hash** (p.ej., SHA-256); el PDF incluye hash/QR; cambios quedan en **auditLogs**.

**13.4 Historias de Usuario (HU)**

Las siguientes Historias de Usuario fueron definidas a partir de los Requisitos Funcionales (RF) y en conjunto con los Stakeholders de **Cosemar SPA**, utilizando la técnica *Product Backlog Building (PBB)*.

Cada HU sigue el formato estándar: **“Como [rol], quiero [acción], para [valor/beneficio]”**.

Estas historias permiten dar trazabilidad entre los **Objetivos (Cap. 12)**, los **Requisitos (Cap. 13)** y el

**Plan de Trabajo (Cap. 14)**.

* **HU-01:** Como operador, quiero registrar un incidente con GPS y descripción, para generar un folio único y trazable.
* **HU-02:** Como operador, quiero adjuntar evidencias (fotos, videos, PDF), para respaldar el incidente con pruebas.
* **HU-03:** Como usuario, quiero autenticarme en el sistema, para acceder de forma segura a mis funcionalidades.
* **HU-04:** Como sistema, quiero persistir incidentes en MongoDB Atlas con folio ECO-YYYYMMDD-######, para garantizar unicidad y trazabilidad.
* **HU-05:** Como supervisor, quiero que el sistema clasifique automáticamente incidentes con IA, para priorizar según severidad.
* **HU-06:** Como operador, quiero subir evidencias a Firebase Storage mediante signed URLs, para resguardar información de forma segura.
* **HU-07:** Como supervisor, quiero visualizar un Dashboard básico con filtros y búsqueda por folio, para revisar incidentes de manera ágil.
* **HU-08:** Como supervisor, quiero recibir notificaciones automáticas (WhatsApp/Email), para reaccionar rápidamente a incidentes críticos.
* **HU-09:** Como auditor, quiero contar con un registro completo de logs y acciones, para asegurar la trazabilidad y cumplimiento normativo.
* **HU-10:** Como supervisor, quiero generar reportes PDF oficiales con sello temporal y QR, para validar y entregar evidencias confiables.
* **HU-11:** Como supervisor, quiero gestionar y reasignar incidentes, para coordinar mejor la atención de casos.
* **HU-12:** Como gerente, quiero un Dashboard avanzado con KPIs y reportería, para evaluar el desempeño del área en tiempo real.
* **HU-13:** Como administrador, quiero gestionar usuarios, roles y parámetros globales, para mantener el control del sistema.
* **HU-14:** Como sistema, quiero manejar integraciones y fallback en notificaciones, para asegurar continuidad operativa.
* **HU-15:** Como administrador, quiero aplicar políticas de retención y archivado automático, para cumplir con normativas legales y de la empresa.

**13.5 Ejemplos de criterios de aceptación (clave)**

**RF-07 Notificaciones automáticas (HIGH)**

* *Given* un incidente clasificado **HIGH**, *When* se confirma su creación, *Then* el sistema envía **WhatsApp** al Supervisor con folio/ubicación y **Email** a Legal con PDF/enlaces, *And* registra el resultado en **auditLogs**.

**RF-10 PDF con sello temporal**

* *Given* un incidente en estado **CLOSED**, *When* el Supervisor confirma el cierre, *Then* el sistema genera un **PDF** con folio, sello de tiempo y QR verificable, *And* lo adjunta al incidente y lo envía por **Email**.

**RF-04 Offline**

* *Given* que no hay conexión, *When* el Operador crea un incidente con evidencia, *Then* el registro queda **en cola local** y se **sincroniza** automáticamente al recuperar red, manteniendo **folio definitivo** y **timestamp** de creación.

**13.6 Trazabilidad mínima (gancho para RTM)**

* RF-01/02/03/04 ↔ HU-01/HU-02 ↔ CP-01/CP-02/CP-03 ↔ Módulo App/API ↔ Estado
* RF-06/07 ↔ HU-05/HU-06 ↔ CP-10/CP-11 ↔ Módulo n8n/Integraciones ↔ Estado
* RF-10/11/09 ↔ HU-08/HU-09 ↔ CP-20/CP-21/CP-22 ↔ Módulo API/Web ↔ Estado

**14. Plan de Trabajo (11 semanas / 4 Sprint’s)**

**14.1 Calendario general**

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Semanas | Enfoque | Meta del incremento |
| Sprint 1 | (Semanas 1–3) | Registro base, offline, API/DB | App registra incidentes con GPS/evidencias mínimas, API persiste con folio único |
| Sprint 2 (MVP) | (Semanas 4–5) | IA + evidencias + Dashboard básico | Clasificación automática con IA, subida de evidencias en Storage y Dashboard inicial |
| Sprint 3 | (Semanas 6–8) | Notificaciones + reportería + KPIs | Alertas WhatsApp/Email, reportería PDF con sello, KPIs y filtros en Dashboard |
| Sprint 4 (Piloto final) | (Semanas 9–11) | RBAC + seguridad/performance + piloto | Roles completos, seguridad optimizada, piloto en terreno, monitoreo y cierre de brechas |

**Definiciones globales:**

* **DoR (Definition of Ready):** historia con usuario/valor, criterios Gherkin, mock/contrato API, datos de prueba, riesgos conocidos.
* **DoD (Definition of Done):** código en main, cobertura de tests ≥80% (API), pruebas de integración/E2E clave, SAST/DAST sin hallazgos críticos, documentación mínima, logging implementado, Deploy en Staging y demo funcional.

**14.2 Sprint 1 (Semanas 1–3) – Registro base + Offline + API/DB**

**Objetivo:** entregar un primer corte funcional de EcoTrack con registro digital y persistencia en base de datos.

**Historias/alcance:**

* HU-01 Registrar incidente (RF-01, RF-03, RF-04).
* HU-02 Adjuntar evidencia (RF-02).
* HU-03 Autenticación básica (login + JWT).
* HU-04 Persistencia en Atlas con folio único.

**Tareas clave:**

* FE Móvil: formulario guiado, GPS/timestamp, cola offline-first con reintentos.
* BE: Endpoint POST /incidents, esquema inicial, índices base, validación payload.
* DevOps: pipeline CI (lint + unit + integration + SAST), despliegue a Staging.
* QA: casos de prueba iniciales (CP-01, CP-02).

**Entregables:**

* App móvil registra incidente con ≥1 evidencia y GPS.
* Persistencia en MongoDB con folio único.
* Swagger inicial en /docs.

**14.3 Sprint 2 (Semanas 4–5) MVP: IA + Evidencias + Dashboard básico**

**Objetivo:** completar el MVP incorporando IA, almacenamiento de evidencias y un Dashboard inicial.

**Historias/alcance:**

* HU-05 Clasificación IA (RF-06).
* HU-06 Evidencias en Firebase Storage (RF-02).
* HU-07 Dashboard básico (lista, filtros mínimos, búsqueda).

**Tareas clave:**

* n8n + OpenAI: clasificación de severidad y resumen automático.
* BE: endpoints de subida de evidencias (signed URL’s).
* FE Web: tabla de incidentes con filtros básicos.
* QA: pruebas de integración, validación de resultados IA.

**Entregables:**

* Clasificación automática de incidentes con severidad visible.
* Evidencias almacenadas en Storage.
* Dashboard funcional con listado inicial.

**14.4 Sprint 3 (Semanas 6–8) – Notificaciones + Reportería + KPIs**

**Objetivo:** habilitar notificaciones inmediatas, reportería con validez probatoria y KPIs de gestión.

**Historias/alcance:**

* HU-08 Notificaciones automáticas (RF-07).
* HU-09 Auditoría y logs inmutables (RF-09).
* HU-10 Reporte PDF con sello temporal y QR (RF-10).
* HU-11 Gestión avanzada Supervisor (RF-08).

**Tareas clave:**

* n8n: flujo de alertas críticas (WhatsApp + Email) con reintentos.
* BE: generación de PDF con hash y sello temporal.
* FE Web: vista Supervisor con reasignación, KPIs iniciales y filtros avanzados.
* QA: pruebas E2E (alta severidad → alerta WA + email).

**Entregables:**

* Notificaciones automáticas de incidentes HIGH.
* PDF’s oficiales con sello temporal y QR.
* Dashboard con KPIs básicos y reportería.

**14.5 Sprint 4 (Semanas 9–11) – RBAC + Seguridad + Piloto final**

**Objetivo:** consolidar el sistema con seguridad, roles completos y validación en un piloto en terreno.

**Historias/alcance:**

* HU-12 Dashboard avanzado con KPIs y reportería (RF-11, RF-12).
* HU-13 Administración de usuarios y parámetros (RF-13).
* HU-14 Integraciones y fallback (RF-15, RF-16).
* HU-15 Políticas de retención y archivado automático (RF-17).

**Tareas clave:**

* BE: implementación RBAC completo (Operador, Supervisor, Legal, Admin).
* DevOps: Hardening de seguridad (p95 < 300 ms, alertas, monitoreo).
* FE Web: KPIs avanzados, exportaciones CSV/XLSX.
* QA: validación de retención, pruebas de carga.
* Piloto en terreno con usuarios reales.

**Entregables:**

* Roles y permisos completos (RBAC).
* Seguridad y performance optimizados.
* Piloto validado con reporte final de brechas.

**15. Evidencias**

**15.1 Estructura y gobernanza de evidencias**

* **Repositorio** (sugerido):
* /docs/rtm.xlsx
* /backlog/sprint-01.xlsx, sprint-02.xlsx, ...
* /qa/casos\_prueba/CP-xx.md (fuente) y /qa/reportes/ (salidas de ejecución)
* /ci/artifacts/{fecha}/ (junit.xml, coverage.html, sast.json, dast.html)
* /evidencias/pdf/ (ECO-YYYYMMDD-######.pdf)
* /evidencias/capturas/ (screenshots dashboard, WA/email)
* **Responsables**:
  + **QA**: RTM, casos de prueba, reportes de ejecución.
  + **DevOps**: reportes CI/CD (JUnit, coverage, SAST/DAST) y retención.
  + **PO**: validación de RTM y relación Req ↔ Evidencia.
* **Versionado**: toda evidencia se versiona por **commit/tag** del sprint; RTM con **consecutivo** (v1.0, v1.1…).
* **Criterio de auditoría**: cada evidencia debe referenciar **Req ID / HU / Sprint / Commit/Tag** y **responsable**.

**15.2 RTM — Matriz de Trazabilidad (/docs/rtm.xlsx)**

**Objetivo**: asegurar trazabilidad **Req ↔ Historia(s) ↔ Caso(s) de Prueba ↔ Módulo ↔ Estado ↔ Evidencia**.

**Columnas mínimas**:

* Req ID (RF/RNF) | Descripción requisito | HU vinculadas | Casos de prueba (CP-xx) | Módulo (App/API/Web/n8n) | Criterio de aceptación | Estado (Pend/En curso/OK) | Evidencia (link) | Observaciones

**Ejemplos**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Req ID | Descripción | HU | CP | Módulo | Criterio | Estado | Evidencia |
| RF-01 | Registrar incidente con GPS y 1 evidencia mínima | HU-01 | CP-01, CP-02 | App/API | Folio único + timestamp | OK | captura app + POST 201 |
| RF-07 | Notificación HIGH por WA/Email en ≤2 min | HU-08 | CP-10, CP-11 | n8n/Integraciones | ≥95% entregas | En curso | logs n8n + messageId |
| RF-10 | PDF con folio, sello y QR al cierre | HU-10 | CP-20 | API/Web | Generación 100% “CLOSED” | OK | PDF ECO-... |
| RNF-02 | p95 < 300ms endpoints críticos | HU-12/13 | CP-30 | API | p95 < 300ms | En curso | reporte perf CI/CD |

**Buenas prácticas**:

* **Mantención viva** por QA (actualización al cierre de cada sprint).
* Enlace directo a **artifacts CI/CD** y a **PDF’s** generados.
* Una pestaña por sprint o un filtro por Sprint.

**15.3 Sprint Backlog (por sprint)**

**Objetivo**: evidenciar **planificación, ejecución y control** del trabajo comprometido.

**Formato mínimo (xlsx o gsheet)**:

* Columnas: ID-HU | Tarea | Responsable | Estimación (h) | Estado | Bloqueos | Evidencia
* Reglas: **tareas ≤ 8h**, cada tarea debe tener **responsable** y **enlace** a evidencia (PR, build, demo).

**Ejemplo (Sprint 1)**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID-HU | Tarea | Responsable | Estim. (h) | Estado | Bloqueos | Evidencia |
| HU-01 | Form móvil + validaciones | FE Móvil | 10 | En curso | — | PR#34, demo.mp4 |
| HU-01 | GPS + timestamp | FE Móvil | 6 | Pendiente | Permisos SO | issue-23 |
| HU-01 | POST /incidents + folio | BE | 8 | Hecho | — | junit.xml, Swagger |
| HU-02 | Adjuntar evidencia (signed URL) | BE | 6 | Hecho | — | PR#37 |
| HU-03 | Login + JWT | BE | 6 | Hecho | — | coverage.html |
| HU-01 | Unit tests | QA | 6 | Pendiente | Datos prueba | CP-01.md |
| CI | Pipeline lint/unit/integration/SAST | DevOps | 6 | Hecho | — | run-2025-09-01 |

**Artefactos anexos**:

* Exportación del **Board** al finalizar el sprint (CSV/PDF).
* **Acta de Retrospectiva** con acciones de mejora y responsables.

**15.4 Casos de prueba QA (/qa/casos\_prueba y /qa/reportes)**

**Objetivo**: validar criterios de aceptación funcionales y no funcionales.

**Plantilla (CP-xx.md)**:

ID: CP-10

Requisito(s): RF-07 (Notificaciones), RNF-20 (Entregabilidad)

HU: HU-08

Tipo: E2E / Funcional

Prioridad: Alta

Precondiciones: Incidente creado con clasificación HIGH (manual o IA)

Datos de prueba: desc="Derrame ácido", zona="Norte", foto=ok.jpg

Pasos:

1. Crear incidente HIGH

2. Confirmar emisión de evento incident.created

3. Verificar WA a Supervisor (template incident\_alert\_high)

4. Verificar Email a Legal con enlace a evidencias y PDF (si aplica)

Resultado esperado:

- WhatsApp entregado (status=delivered) ≤ 2 min

- Email entregado (messageId) ≤ 2 min

- AuditLogs registra envío/resultado y correlationId

Evidencias: /qa/reportes/CP-10\_{fecha}/capturas/, logs\_n8n.json

**Ejemplos de CP clave**:

* **CP-01 (RF-01)**: crear incidente con GPS + 1 foto → **201**, folio ECO-..., visible en dashboard.
* **CP-11 (RF-07 + RNF-20)**: **≥95%** alertas críticas entregadas en ≤2 min (medición lote 20 eventos).
* **CP-20 (RF-10)**: cierre genera **PDF con sello y QR**; verificación de hash/QR correcta.
* **CP-30 (RNF-02)**: p95 **< 300 ms** en POST /incidents, GET /incidents?filters, PATCH /incidents/:id.

**Ejecución y reporte**:

* Al final de cada sprint: **matriz CP vs. Resultado (OK/Fail/NA)** con enlaces a capturas, logs y artifacts CI.

**15.5 Reportes de CI/CD (/ci/artifacts/{fecha})**

**Objetivo**: demostrar calidad técnica continua (automatizada).

**Incluye**:

* **Pruebas**: junit.xml (unit/integration), **coverage.html** (objetivo API ≥ 80%).
* **Seguridad**: **SAST** (CodeQL/semgrep, sast.json), **DAST** (scan básico Staging, dast.html).
* **Contratos**: validación de **OpenAPI** (tests de contrato).
* **Performance**: resultados sintéticos (latencias p50/p95/p99), al menos endpoints críticos.
* **Deploy logs**: evidencia de despliegue a Staging/Prod, *approval gates*, *smoke tests*.

**Quality Gates (no merge)**:

* Lint sin errores bloqueantes.
* **Coverage API ≥80%**.
* SAST: **0 críticos** (Crit) y **0 altos** (High) abiertos.
* E2E clave **OK** en Staging (para features marcadas “Hechas”).

**Retención**:

* Guardar artifacts **por sprint** y **por release** (tag): vS2.0, vS3.0, etc.

**15.6 PDF’s con folio y sello (/evidencias/pdf)**

**Objetivo**: dejar evidencia **probatoria y auditable** del cierre del incidente.

**Convención de nombre**:

* ECO-YYYYMMDD-######.pdf (ej.: ECO-20250901-000123.pdf).

**Contenido mínimo del PDF**:

* **Folio**, fechas (**creación/asignación/cierre**), tipo, severidad, descripción.
* Ubicación (coordenadas y mapa miniatura).
* Responsables (**reportedBy**, **assignedTo**).
* **Evidencias** (embebidas o enlazadas con **signed URL’s**).
* **Bitácora resumida** (extracto **auditLogs** relevantes).
* **Sello temporal** (UTC), **hash** del documento y **QR** con URL de verificación.

**Metadatos asociados (ejemplo JSON)**:

{

"folio": "ECO-20250901-000123",

"issuedAt": "2025-09-01T18:22:31Z",

"hash": "SHA256:…",

"verifyUrl": "https://app.ecotrack/incident/ECO-20250901-000123",

"generatedBy": "reports@ecotrack"

}

**Verificación**:

* Endpoint /reports/verify?folio=... (opcional) devuelve hash/fecha.
* Registro de generación y envío en **auditLogs** y **logs de n8n** (para Email).

**15.7 Cómo se valida la rúbrica con estas evidencias**

* **Indicadores de calidad (punto 5 de la 1.4)**:
  + Producto: TTR/TTA, % evidencias completas, uptime, % alertas ≤2 min → **Dashboard KPIs**, logs n8n, PDF’s.
  + Proceso: velocidad, Burndown, defectos, DoD → **Sprint Backlog**, **CI/CD**, **actas de retro**.
  + DevOps (DORA): deploy/sprint, MTTR, change fail rate → **artifacts CI/CD** y tableros.

**15.8 Checklist de cierre por sprint (entregable mínimo)**

* ✅ **RTM** actualizado y subido a /docs/rtm.xlsx.
* ✅ **Sprint Backlog** exportado (CSV/PDF) con estados finales.
* ✅ **Paquete CI/CD**: junit, coverage (≥80%), SAST/DAST, logs de deploy.
* ✅ **Casos de prueba** ejecutados con reporte y capturas en /qa/reportes/.
* ✅ **PDF’s** generados para incidentes **CLOSED** del sprint con hash/QR.
* ✅ **Acta de retrospectiva** con acciones para el siguiente sprint.

**16. Indicadores de Calidad**

**16.1 Propósito y alcance**

Definir **qué medimos**, **cómo lo calculamos**, **dónde obtenemos los datos** y **cuándo se evalúan** para asegurar que EcoTrack cumple estándares de **producto, proceso y DevOps**. Todos los indicadores se reportan en **cada Sprint Review** y se consolidan al cierre de la **Fase 1**.

**16.2 Indicadores de Producto (servicio/uso)**

**Metas Fase 1:** TTR ↓60%, TTA ↓60%, evidencias completas ≥95%, uptime ≥98%.

**16.2.1 TTA — Time To Assign (↓60%)**

* **Definición:** tiempo desde la creación hasta la **asignación** del incidente.  
  **Fórmula:** TTA = assignedAt – createdAt
* **Cálculo:** mediana (**p50**) y p90 por sprint; comparación contra **línea base** (promedio histórico o semana 1).
* **Fuente:** colección incidents (MongoDB Atlas).
* **Meta:** **reducción ≥ 60%** del p50 TTA respecto línea base al finalizar Sprint 5.
* **Responsable:** **PO** (valor), **BE** (campos tiempo), **QA** (validación).

**16.2.2 TTR — Time To Resolve (↓60%)**

* **Definición:** tiempo desde la creación hasta el **cierre** del incidente.  
  **Fórmula:** TTR = closedAt – createdAt
* **Cálculo:** p50/p90 por sprint; segmentado por severidad.
* **Fuente:** incidents.timestamps.
* **Meta:** **reducción ≥ 60%** del p50 TTR vs. línea base al Sprint 5.
* **Responsable:** **Supervisor/PO** (flujo), **QA** (métrica).

**Ejemplo (MongoDB Aggregation):**

* Mediana/p90 de TTR en el sprint actual.
* Filtro por fechas (createdAt∈[inicioSprint, finSprint]) y status="CLOSED".

**16.2.3 Evidencias completas (≥95%)**

* **Definición:** incidente con **mínimo 1 foto + descripción + GPS válido**.
* **Fórmula:**  
  EvidenciaCompleta = (evidenceCount ≥1) ∧ (description.length ≥ N) ∧ (location.precision ≤ P)  
  %Completos = (# incidentes con EvidenciaCompleta / # incidentes creados) × 100
* **Cálculo:** porcentaje por sprint y por zona.
* **Fuente:** incidents (campos evidence, description, location).
* **Meta:** **≥ 95%**.
* **Responsable:** **FE Móvil** (captura/validaciones), **QA** (auditoría de muestreo).

**16.2.4 Uptime (≥98%)**

* **Definición:** disponibilidad mensual de la API/dashboards.
* **Fórmula:** Uptime = 1 – (Downtime/TotalTime)

Con **TotalTime** en el período mensual; **Downtime** se registra cuando los healthchecks fallan (>30s a 1min).

* **Fuentes:**
  + Servicio de **healthcheck** (cada 60s).
  + Logs de balanceador/monitor (p. ej., Firebase/ingress).
* **Meta:** **≥ 98%** (Fase 1).
* **Error budget mensual:** 2% ≈ **14,4 h** en 30 días.
* **Responsable:** **DevOps** (monitoreo/alertas) y **TL** (mitigaciones).

**Alerta:** si el consumo del **error Budget** supera el 50% del período, activar **freeze parcial** (solo hotfix) y **plan de resiliencia**.

**16.3 Indicadores de Proceso (Agile)**

**Metas Fase 1:** ritmo sostenible, cumplimiento DoD y flujo estable.

**16.3.1 Velocidad del equipo**

* **Definición:** **puntos de historia** entregados “Done” por sprint.
* **Uso:** proyección de alcance y capacidad; no comparativo entre equipos.
* **Fuente:** tablero (Jira/Trello/planilla).
* **Meta:** estabilización a partir de Sprint 2.
* **Responsable:** **SM/PO**.

**16.3.2 Burndown Chart**

* **Definición:** trabajo pendiente vs. tiempo del sprint.
* **Criterio:** curva con tendencia decreciente; desvíos se explican en Review.
* **Fuente:** board / export CSV.
* **Responsable:** **SM** (seguimiento diario).

**16.3.3 DoD cumplido (tests ≥80%)**

* **Definición:** % de historias “Done” que cumplen **todos los criterios DoD** (tests, docs, seguridad, Deploy a Staging).
* **Fórmula:**  
  %DoD = (# historias con DoD cumplido / # historias Done) × 100
* **Cobertura API:** **≥ 80%** (Unit) obligatoria por sprint.
* **Fuentes:** reportes **CI/CD** (Jest Coverage), acta de Review.
* **Meta:** **100%** historias con DoD, **Coverage ≥80%**.
* **Responsable:** **QA/DevOps**.

**Quality Gate:** si Coverage <80% o SAST con hallazgos críticos → **no merge**.

**16.4 Indicadores DevOps (DORA)**

**Metas Fase 1:** Deploy ≥1/sprint, MTTR < 4h.

**16.4.1 Frecuencia de despliegue**

* **Definición:** despliegues a **Staging/Producción** por sprint.
* **Meta:** **≥ 1** a Producción por sprint (al menos 1 release funcional).
* **Fuente:** logs de **GitHub Actions** / registros de release.
* **Responsable:** **DevOps**.

**16.4.2 MTTR — Mean Time To Restore (<4h)**

* **Definición:** tiempo medio para restaurar el servicio tras un incidente en Prod.
* **Fórmula:** MTTR = avg(restoreAt – incidentStart)
* **Fuente:** canal de incidentes + logs (marcas de tiempo).
* **Meta:** **< 4 horas**.
* **Responsable:** **DevOps/TL**.

**16.4.3 Change Failure Rate (≤15%) *(recomendado)***

* **Definición:** % de despliegues que causan incidentes/hotfixes.
* **Fórmula:** CFR = (# despliegues fallidos / # despliegues totales) × 100
* **Meta:** **≤ 15%**.
* **Fuente:** issues de post-deploy/hotfix + registro de releases.
* **Responsable:** **DevOps/SM**.

**Lead time for changes** (opcional): tiempo desde merge a despliegue; objetivo ≤ 1 día a Staging.

**16.5 Captura de datos e implementación**

* **Producto (TTR/TTA/evidencias):** MongoDB Atlas (incidents, evidence, timestamps).
* **Uptime:** healthchecks y logs de balanceador/hosting.
* **Proceso (velocidad/Burndown/DoD):** tablero y artefactos de sprint; Coverage CI.
* **DORA:** logs de **GitHub Actions**, etiquetado de releases, issues de hotfix.
* **Observabilidad:** logs estructurados con correlationId; panel técnico (latencias p95, 5xx) y panel de negocio (TTR/TTA/% evidencias).

**16.6 Reportes y tableros**

* **Dashboard de KPIs (Producto):** TTR/TTA p50/p90, % evidencias completas, uptime mensual.
* **Panel de Proceso:** velocidad por sprint, Burndown, % DoD, defectos.
* **Panel DevOps:** deploys por sprint, MTTR, CFR, Coverage, SAST/DAST.
* **Cadencia:** actualización **continua** y revisión en **cada Sprint Review**.
* **Exportables:** CSV/XLSX y PDF para anexar al informe.

**16.7 Umbrales, alertas y acciones**

* **Uptime < 98% (mes):** activar **plan de resiliencia**, revisar error Budget, priorizar estabilidad en backlog.
* **p95 > 300 ms (endpoints críticos):** optimizar índices/queries, cache liviana, revisar n+1.
* **% Evidencias < 95%:** reforzar validaciones móviles, Checklist UX, capacitación.
* **Coverage < 80% / SAST crítico:** **bloqueo de merge**, hotfix de seguridad.
* **Deploy < 1/sprint o MTTR ≥ 4h:** ajustar **pipeline**, runbooks y simulacros de recuperación.

**16.8 Evidencias asociadas (cómo se demuestra)**

* **Capturas de Dashboard** (KPIs de producto).
* **Export Burndown/velocidad** por sprint.
* **Artifacts CI/CD**: junit.xml, coverage.html, sast.json, dast.html.
* **Log de releases** y **actas de postmortem** (si aplica).
* **Anexos PDF** con folio y sello temporal (muestran consistencia con KPIs).

**17. Cronograma**

**Calendario de referencia:** 01-09-2025 → 16-11-2025

**Sprint’s:**

* **S1 (Sem 1–3)** — Registro base, offline, API/DB.
* **S2 (Sem 4–5, MVP)** — IA + evidencias + Dashboard básico.
* **S3 (Sem 6–8)** — Notificaciones + auditoría.
* **S4 (Sem 9–11, cierre)** — PDF + KPIs + RBAC, seguridad/perf y piloto.

**Artefactos transversales por sprint:** Product/Sprint Backlog, RTM, tablero (Scrum), CI/CD (lint, tests, coverage), actas de Review y Retro, reporte de riesgo actualizado.

**17.1 Vista general por sprint (resumen ejecutivo)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint | Semanas / Fechas | Objetivo de valor | Incremento esperado | Gates de calidad (salida) |
| S1 | **1–3** (01–21 Sep) | Registrar incidentes con evidencias mínimas, base API/DB y offline | App móvil con formulario+GPS+timestamp, API POST /incidents, Mongo Atlas con esquema, sincronización offline mínima | CI con Lint+Unit (API) **≥80%**, Swagger inicial, Demo “creación de incidente con folio” en Staging |
| S2 (MVP) | **4–5** (22 Sep–05 Oct) | Clasificación IA + evidencias seguras + Dashboard mínimo | Flujo n8n incident.created → OpenAI (severity, summary), evidencias via Signed URL, Dashboard con lista/filtros básicos | E2E smoke (crear→clasificar→ver), RTM actualizado, Review de MVP con stakeholders |
| S3 | **6–8** (06–26 Oct) | Notificaciones en tiempo real + auditoría trazable | WhatsApp/Email con plantillas, thresholds de entrega, auditLogs visibles en detalle de incidente | Métrica de entregabilidad **≥95% ≤2 min**, visor de auditLogs, E2E inicial en Staging |
| S4 (cierre) | **9–11** (27 Oct–16 Nov) | Reporte PDF sellado + KPIs + RBAC + hardening + piloto | PDF con sello/QR, KPIs (TTA/TTR/% evidencias), RBAC completo, p95<300 ms, monitoreo y piloto | DAST Staging OK, panel de latencias/errores, acta de readiness, acta de cierre/piloto |

**17.2 Desglose semana por semana**

**Formato por semana:** Objetivo → Actividades clave → Gates QA/DevOps → Entregables/Evidencias → Riesgos & mitigación → Responsables foco.

**Semana 1 (01–07 Sep.) — S1**

* **Objetivo:** Arranque técnico y primer registro end-to-end.
* **Actividades:**  
  App: formulario, validaciones, GPS/timestamp • API POST /incidents • Esquema incidents en Atlas (índices base) • JWT auth básica.
* **Gates:** Lint+Unit (API) ≥60% (rampa S1) • Swagger /docs publicado.
* **Entregables:** Demo crear incidente (201 con folio) • Swagger v0 • RTM v0 (RF-01/02/03 ↔ HU-01/02).
* **Riesgos/mitigación:** Permisos GPS (guía in-app) • Duplicados (idempotency key).
* **Responsables foco:** FE Móvil, BE, TL, QA.

**Semana 2 (08–14 Sep.) — S1**

* **Objetivo:** **Offline-first** + evidencia mínima.
* **Actividades:**  
  Cola de sincronización (reintentos) • Adjuntar foto → Signed URL (stub) • Índices severidad/fecha • Unit tests API.
* **Gates:** Coverage API **≥70%** • PR con SAST básico (CodeQL/semgrep).
* **Entregables:** HU-01/02 “Done” • RTM v1 con RF-01/02/04 • Reporte CI.
* **Riesgos/mitigación:** Conectividad irregular → pruebas modo avión + reintentos exponenciales.
* **Responsables:** FE Móvil, BE, QA, DevOps.

**Semana 3 (15–21 Sep.) — S1**

* **Objetivo:** Estabilizar base para IA/evidencias.
* **Actividades:**  
  Endpoint POST /incidents/:id/evidence • Normalizar metadatos (hash/ts) • Hardening básico CORS/rate-limit.
* **Gates:** Coverage API **≥80%** • Smoke Staging (crear→adjuntar→listar).
* **Entregables:** Demo E2E corto • Acta Review S1 • Backlog S2 afinado.
* **Riesgos/mitigación:** Tamaño de archivos → límites y feedback UI.
* **Responsables:** BE, FE Móvil, QA.

**Semana 4 (22–28 Sep.) — S2 (MVP)**

* **Objetivo:** IA (clasificación parte 1) + soporte de evidencias.
* **Actividades:**  
  n8n Webhook incident.created • OpenAI (JSON {severity, summary}) • Persistencia de resultado IA • Visor de evidencia por Signed URL.
* **Gates:** E2E Staging (crear→clasificar→ver) • Logs con correlationId.
* **Entregables:** Flujo IA operativo • CP integración IA.
* **Riesgos/mitigación:** Costo IA → cache por folio • Privacidad → truncar PII.
* **Responsables:** n8n, BE, QA.

**Semana 5 (29 Sep–05 Oct) — S2 (MVP)**

* **Objetivo:** Cerrar MVP con Dashboard básico.
* **Actividades:**  
  Dashboard lista/filtros • Severidad reclasificable por Supervisor • Ajustes IA (thresholds) • UX básicos.
* **Gates:** Review S2 con stakeholders • RTM actualizado • E2E mínimo Web.
* **Entregables:** **MVP**: registro+evidencias+IA+dashboard • Reporte CI/CD.
* **Riesgos/mitigación:** Carga lista → paginación y debounced filters.
* **Responsables:** FE Web, BE, TL, QA.

**Semana 6 (06–12 Oct) — S3**

* **Objetivo:** Notificaciones (parte 1).
* **Actividades:**  
  n8n reglas HIGH → WhatsApp/Email • Plantillas aprobadas • Métrica de entregabilidad.
* **Gates:** Pruebas de envío en lote controlado • Tiempos de entrega medidos.
* **Entregables:** Entregabilidad **≥90% ≤3 min** (escala inicial) • Logs audit envío.
* **Riesgos/mitigación:** Falla WA → fallback SMTP; reintentos n8n.
* **Responsables:** n8n, DevOps, QA.

**Semana 7 (13–19 Oct) — S3**

* **Objetivo:** Auditoría y trazabilidad.
* **Actividades:**  
  auditLogs completo (who/role/when/from→to/ip/hash) • Visor en detalle de incidente • E2E inicial (creación→notificación→auditoría).
* **Gates:** E2E pasa en Staging • SAST/DAST sin críticos.
* **Entregables:** AuditLogs consultables • CP-20 (hash/QR preparado).
* **Riesgos/mitigación:** Ruido en logs → niveles, sampling, PII redaction.
* **Responsables:** BE, FE Web, QA.

**Semana 8 (20–26 Oct) — S3**

* **Objetivo:** Consolidar S3 y preparar S4.
* **Actividades:**  
  Filtros avanzados • Exportaciones CSV/XLSX • Tuning consultas (índices) • Retro de proceso.
* **Gates:** Entregabilidad **≥95% ≤2 min** • Review S3.
* **Entregables:** S3 cerrado con métricas • RTM con RF-07/09/12/16.
* **Riesgos/mitigación:** Latencias → índices / paginación server-side.
* **Responsables:** FE Web, BE, TL.

**Semana 9 (27 Oct–02 Nov) — S4**

* **Objetivo:** PDF con sello (parte 1) + RBAC (parte 1).
* **Actividades:**  
  Servicio PDF (folio, sello, QR, evidencias) • Endpoint descarga • Roles (Operador/Supervisor/Legal/Admin) en rutas críticas.
* **Gates:** Generación PDF en Staging • Pruebas de autorización básicas.
* **Entregables:** Prototipo PDF • RBAC preliminar • CP PDF.
* **Riesgos/mitigación:** Peso PDF → compresión/miniaturas.
* **Responsables:** BE, QA.

**Semana 10 (03–09 Nov) — S4**

* **Objetivo:** KPIs + hardening/monitoreo.
* **Actividades:**  
  KPIs (TTA/TTR/% evidencias) • Panel técnico (latencias p95, errores) • Alertas/healthchecks • Rate-limit, CSP/CORS.
* **Gates:** DAST Staging OK • p95 preliminar **< 350 ms** (ajuste final semana 11).
* **Entregables:** Dashboard KPIs • Tablero técnico • Acta readiness preliminar.
* **Riesgos/mitigación:** Spikes de carga → escalado horizontal API.
* **Responsables:** DevOps, BE, TL.

**Semana 11 (10–16 Nov) — S4**

* **Objetivo:** Piloto y cierre.
* **Actividades:**  
  Feature flags ON (WA/IA) en Prod controlada • Onboarding, soporte y correcciones • Afinar p95 **< 300 ms** en endpoints críticos.
* **Gates:** Checklist de readiness → OK • Post-deploy checks (errores/latencias) • Review final.
* **Entregables:** Piloto ejecutado • Acta de cierre & lecciones aprendidas • Métricas de impacto (TTA/TTR↓≥60%, % evidencias ≥95%).
* **Riesgos/mitigación:** Incidencias en producción → rollback/canary, runbooks.
* **Responsables:** Equipo completo.

**17.3 Hitos del cronograma**

* **H1 – Fin S2 (Semana 5):** **MVP** operativo (registro + evidencias + IA + Dashboard).
* **H2 – Fin S3 (Semana 8):** Notificaciones WA/Email **≥95% ≤2 min** + auditoría visible.
* **H3 – Fin S4 (Semana 11):** PDF sellado + KPIs + RBAC + **piloto validado** con p95<300 ms.

**17.4 Ruta crítica y dependencias**

* **Ruta crítica:** Registro/Offline (S1) → IA+Evidencias (S2) → Notificaciones+Auditoría (S3) → PDF+KPIs+RBAC/Piloto (S4).
* **Dependencias clave:**
  + IA/Evidencias requieren API/DB base (**S1**).
  + Notificaciones requieren IA estable y eventos (**S2**).
  + PDF requiere modelo estable y auditLogs (**S3**).
  + Piloto depende de RBAC, KPIs y hardening (**S4**).

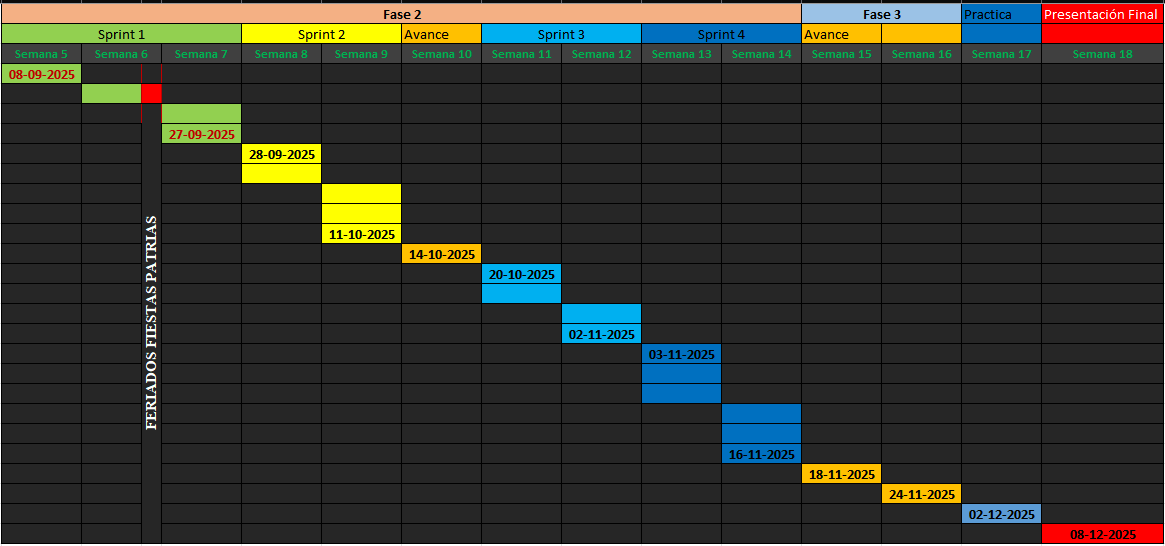
**17.5 Métricas y criterios de aceptación por sprint**

* **S1:** Coverage API **≥80%**; flujo crear→adjuntar→listar estable en Staging.
* **S2:** E2E MVP (crear→clasificar→ver); Dashboard mínimo funcional; RTM actualizado.
* **S3:** Entregabilidad **≥95% ≤2 min**; visor de auditLogs; E2E notificaciones+audit.
* **S4:** PDF válido y descargable; KPIs operativos; RBAC completo; **p95<300 ms**; piloto ejecutado y documentado.

**17.6 Evidencias académicas a entregar (por sprint)**

* **Todos:** Sprint Backlog, reporte CI/CD (coverage, SAST/DAST), RTM actualizado, actas Review/Retro.
* **S2:** Video demo de MVP.
* **S3:** Capturas/Logs de notificaciones y auditoría; exportaciones CSV/XLSX.
* **S4:** Muestra de PDF sellado (folio/QR), tablero de KPIs, panel de monitoreo, **Acta de Cierre** con métricas de impacto.

****

****

**18. Conclusión General**

**Factibilidad técnica.** La arquitectura de EcoTrack - React (móvil + web), Node.js/Express, MongoDB Atlas, Firebase Storage/Hosting y n8n - está probada en contextos cercanos a producción y se sustenta en prácticas modernas de DevSecOps (CI/CD con GitHub Actions, SAST/DAST, cobertura de pruebas ≥80%). La solución cumple los objetivos no funcionales definidos para el piloto (disponibilidad ≥98%, p95 <300 ms en endpoints críticos, ≥95% de entrega de alertas críticas en ≤2 minutos) e implementa seguridad desde el diseño (JWT/OAuth2, RBAC, URL’s firmadas para evidencias, auditoría, TLS en tránsito y cifrado en reposo). Los riesgos (conectividad intermitente, costos de APIs, adopción) se mitigan con enfoque **offline-first**, límites y tope de consumo, *Caching* y *fallbacks* (p. ej., correo si falla WhatsApp) y un plan de habilitación estructurado.

**Factibilidad económica.** El costo operativo mensual es contenido (≈ CLP 650.000 para un piloto de baja/media escala) gracias al uso de servicios gestionados. El análisis costo/beneficio es favorable por la reducción medible de TTA/TTR (meta ≥60%), el aumento de la completitud de evidencias (≥95%) y los ahorros derivados de una mejor conformidad regulatoria y resolución más rápida de incidentes. El despliegue por etapas (MVP en Sprint 2, piloto en Sprint 5) limita la exposición financiera y valida valor temprano.

**Factibilidad legal.** La plataforma implementa consentimiento informado y limitación de propósito, políticas de retención (3–5 años para incidentes críticos), cadena de custodia mediante **audit logs** inmutables y PDF’s verificables criptográficamente (folio, sello temporal, hash/QR). Los datos se protegen con TLS, cifrado en reposo, mínimo privilegio y controles de privacidad (evitando compartir PII innecesaria con proveedores externos). Bajo estos controles, la evaluación legal es favorable para operar en contextos ambientales regulados.

**Valor para la industria e impacto.** Para servicios de retiro de residuos (ej., Cosemar SPA), EcoTrack ofrece trazabilidad digital de extremo a extremo - desde la captura en terreno hasta el cierre auditable - acelerando la respuesta operativa, elevando la preparación para auditorías y habilitando gestión por KPIs (TTR, TTA, % evidencias, reincidencias). La arquitectura y el flujo se integran de forma natural con herramientas y procesos existentes (correo, WhatsApp, tableros) y entregan una base sólida para la mejora continua (observabilidad, métricas DORA, ADR’s).

**Alineación con competencias del perfil de egreso.** El proyecto integra de forma demostrable: ingeniería de software y gestión ágil (Scrum + PBB), desarrollo seguro de Backend y gestión de datos (Node.js/MongoDB, RBAC, logging), ingeniería de aplicaciones móviles y web (React + Capacitor / React SPA), automatización e integraciones (n8n, OpenAI, WhatsApp, SMTP) y DevOps/calidad (CI/CD, Testing, SAST/DAST, métricas). Esta alineación evidencia la capacidad del egresado para diseñar, implementar, asegurar y operar una solución real centrada en valor dentro de los plazos académicos (11 semanas, 4 Sprint’s).

**19. Reflexiones individuales**

**✦ Carlos Navarro Joansen — PO, SM, QA, DevOps**

Al asumir múltiples roles en EcoTrack, me enfrento al desafío más completo de mi formación. Ser **Product Owner** y **Scrum Master** me exige aprender a escuchar, priorizar y coordinar, manteniendo siempre el norte del proyecto. Como **QA y DevOps**, me toca velar por la calidad y la estabilidad técnica, asegurando que cada entrega esté respaldada por pruebas y automatización.

Este proyecto me conecta con mis intereses más profundos: la **inteligencia artificial**, la **nube** y la **ciberseguridad**. La IA me inspira por su capacidad de transformar datos en decisiones; la nube me permite pensar en soluciones escalables y accesibles; y la seguridad me recuerda que ninguna innovación es completa sin confianza y resguardo.

Mi mayor motivación es demostrar que, con disciplina y pasión, se puede liderar un equipo hacia un objetivo que trascienda lo académico, y que impacte en la sociedad. EcoTrack no es solo un prototipo: es el reflejo de lo que quiero construir en mi futuro profesional, con un enfoque en innovación, responsabilidad y valor real.

**✦ José Almontes Fuenzalida — Tech Lead, BE, FE Web/Móvil**

En EcoTrack, mi papel como **Tech Lead** y responsable del **Backend y Frontend** representa una oportunidad de oro para consolidar mi capacidad técnica. La responsabilidad de guiar las decisiones de arquitectura y transformar los requerimientos en una solución funcional me desafía a pensar siempre en calidad, escalabilidad y simplicidad.

Trabajar con tecnologías como **Node.js, MongoDB Atlas y React** me permite fortalecer mi perfil de desarrollador full Stack, mientras que el liderazgo técnico me impulsa a ser un referente en buenas prácticas y estándares de seguridad. Mi meta es que cada decisión que tomemos deje una base sólida para el crecimiento del sistema y que, al mirar hacia atrás, podamos sentir orgullo de la calidad de lo que construimos.

Me proyecta hacia el futuro imaginar que EcoTrack no solo será un entregable académico, sino un paso clave en mi camino profesional. Sé que las habilidades que desarrolle aquí, desde la programación hasta la mentoría técnica, serán un pilar en mi crecimiento como ingeniero en informática.

**✦ Giovanni Vargas Contreras — FE Web, QA**

Mi rol en EcoTrack, enfocado en el **Frontend web y pruebas de calidad**, me permite aportar en dos áreas que considero esenciales: la experiencia de usuario y la verificación de que todo funcione como se espera. El **Dashboard web** es la ventana a toda la trazabilidad de incidentes, y sé que mi responsabilidad es asegurar que sea claro, eficiente y que entregue valor real a los supervisores y áreas de control.

Al mismo tiempo, participar en las actividades de **QA** me conecta con el compromiso de garantizar que cada funcionalidad tenga evidencias de calidad, desde pruebas unitarias hasta la validación de integraciones. Es un rol que me obliga a ser meticuloso, crítico y colaborativo, cualidades que quiero seguir fortaleciendo.

Pienso en el futuro y me emociona ver cómo esta experiencia marcará mi trayectoria profesional. Aprender a diseñar interfaces útiles y a validar software de manera rigurosa me da la confianza de que puedo crecer en áreas donde convergen **tecnología, diseño y calidad**, siempre con el objetivo de crear soluciones confiables y significativas.

**20. Bibliografía (formato APA 7.ª ed.)**

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum. The Rules of the Game*. Scrum.org. [https://scrumguides.org](https://scrumguides.org?utm_source=chatgpt.com)

MongoDB, Inc. (s. f.). *MongoDB Atlas documentation*. MongoDB. Recuperado el 2 de septiembre de 2025, de [https://www.mongodb.com/docs/atlas/](https://www.mongodb.com/docs/atlas/?utm_source=chatgpt.com)

Google LLC. (s. f.). *Firebase documentation*. Google. Recuperado el 2 de septiembre de 2025, de [https://firebase.google.com/docs](https://firebase.google.com/docs?utm_source=chatgpt.com)

OpenAI. (s. f.). *OpenAI API documentation*. OpenAI. Recuperado el 2 de septiembre de 2025, de [https://platform.openai.com/docs](https://platform.openai.com/docs?utm_source=chatgpt.com)

Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (7th ed.). Project Management Institute. ISBN: 9781628256642. https://doi.org/10.1002/9781628256642

**21. Anexos**

Este apartado define **formatos, ejemplos y convenciones** para adjuntar evidencia del proyecto. Los anexos se entregan junto al informe en una carpeta raíz ANEXOS\_ECOTRACK/ con subcarpetas estandarizadas.

**21.1 RTM de ejemplo (/docs/rtm.xlsx)**

**Objetivo:**

asegurar trazabilidad **Req ↔ Historia(s) ↔ Caso(s) de prueba ↔ Módulo ↔ Evidencia ↔ Estado**.

**Campos mínimos (por fila):**

Req ID | Descripción | HU vinculadas | CP (casos de prueba) | Módulo | Criterio de aceptación | Estado | Evidencia (enlace) | Observaciones

**Muestra (extracto):**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Req ID | Descripción | HU | CP | Módulo | Criterio de aceptación | Estado | Evidencia | Observaciones |
| RF-01 | Registrar incidente con GPS y ≥1 evidencia | HU-01 | CP-01, CP-02 | App / API | Folio único + timestamp | **OK** | capturas/CP-01.png, ci/junit.xml | Validado en Staging |
| RF-06 | Clasificación IA (LOW/MED/HIGH) | HU-05 | CP-10 | n8n / OpenAI | Severidad visible y editable por Supervisor | **OK** | logs/n8n\_run\_2025-09-22.json | Cache por folio habilitada |
| RF-07 | Notificación crítica WA/Email ≤2 min | HU-08 | CP-11 | n8n / Integraciones | ≥95% entregas en tiempo | **En curso** | capturas/WA\_template.png | Falta lote de 20 pruebas |
| RF-10 | PDF con folio, sello, QR al cierre | HU-10 | CP-20 | API / Web | 100% de cierres generan PDF | **OK** | pdf/ECO-20250901-000123.pdf | Hash verificado |
| RNF-02 | p95 < 300 ms en endpoints críticos | HU-13 | CP-30 | API | Medido en CI/CD | **En curso** | ci/perf\_p95.html | Optimizar índices |

**21.2 Sprint Backlog inicial (Sprint 1)**

Archivo sugerido: /backlog/sprint-01.xlsx

**Columnas obligatorias:**

ID-HU | Tarea | Responsable | Estimación (h) | Estado | Bloqueos | Evidencia

**Muestra (extracto):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID-HU | Tarea | Responsable | Estim. (h) | Estado | Bloqueos | Evidencia |
| HU-01 | Form móvil + validaciones | FE Móvil | 10 | En curso | — | PR#34, demo/registro.mov |
| HU-01 | GPS + timestamp | FE Móvil | 6 | Pendiente | Permisos SO | issue-23 |
| HU-01 | POST /incidents + folio | BE | 8 | Hecho | — | ci/junit.xml, docs/swagger.html |
| HU-02 | Adjuntar evidencia (signed URL) | BE | 6 | Hecho | — | PR#37 |
| HU-03 | Login + JWT | BE | 6 | Hecho | — | Coverage/index.html |
| HU-01 | Unit tests App/API | QA | 6 | Pendiente | Datos de prueba | qa/CP-01.md |
| CI | Pipeline lint/Unit/Integration/SAST | DevOps | 6 | Hecho | — | ci/run\_2025-09-01/ |

**Regla:** tareas ≤ **8 h**, cada fila con **responsable** y **enlace a evidencia** (PR, build, demo).

**21.3 Capturas de pipeline CI/CD y Dashboard de KPIs**

Carpeta sugerida: /evidencias/capturas/

**Paquete mínimo por sprint:**

1. **CI/CD (GitHub Actions)**
   * ci/junit.xml (pruebas Unit/Integration)
   * ci/Coverage/index.html (**Coverage API ≥80%**)
   * ci/sast.json (SAST sin hallazgos críticos)
   * ci/dast.html (scan básico a Staging)
   * ci/deploy\_logs.txt (Staging/Prod, *approval gates*)
2. **Performance**
   * ci/perf\_p95.html (p50/p95/p99 de endpoints: crear/listar/clasificar/notificar)
3. **Métricas de producto (Dashboard KPIs)**
   * Captura **TTR/TTA** (p50/p90) por sprint
   * **% evidencias completas**
   * **Uptime mensual** (≥98%)
   * Export **CSV/XLSX** desde Dashboard: kpi\_export\_Sx.csv
4. **Integraciones**
   * Capturas de **WhatsApp Cloud API** (entrega: *delivered*, *timestamp*)
   * Captura de **Email SMTP** (messageId, cabeceras)
   * logs/n8n\_run\_YYYYMMDD.json (correlationId y resultado de entregas)