UTE FORENSIA THOT

F1.3.2 Arquitectura de Interoperabilidad (DAI)

THOT

Periodo de Informe 30/09/2025 a 27/02/2026

Fecha:	27/02/2026
Versión:	2.0

Información de control del documento

Descripción	Valor
Título del Documento:	Documento de Arquitectura de Interoperabilidad (DAI)
Nombre del Proyecto:	THOT
Autor del documento:	Sergio Zaera Mata, Sergio Queraltó Pereira, Jaime Castro Cernadas
Propietario del Proyecto:	UTE FORENSIA THOT
Director del Proyecto:	Roberto Gómez-Espinosa
Versión Doc.:	2.0
Confidencialidad:	Media
Fecha:	27/02/2026

Aprobación y revisión del documento:

NOTA: Se requieren todas las aprobaciones. Se deben mantener registros de cada aprobación.

Todos los revisores de la lista se consideran necesarios a menos que se indique explícitamente como Opcionales.

Nombre	Rol	Acción	Fecha
Sergio Zaera Mata	Jefe de Proyecto	Revisa	26/01/2026

Historial de documentos:

El Autor del Documento está autorizado a hacer los siguientes tipos de cambios al documento sin requerir que el documento sea aprobado nuevamente:

- Editorial*, formateo y ortografía.*
- Aclaración.

Para solicitar un cambio en este documento, póngase en contacto con el Autor o el Propietario del Documento.

Las modificaciones de este documento se resumen en la siguiente tabla en orden cronológico inverso (primero la última versión).

Revisión	Fecha	Creada por	Breve descripción de los cambios
0.0	07/10/25	Sergio Zaera Mata	Preparación ToC
0.1	03/11/25	Sergio Queraltó, Jaime Castro	Contribuciones técnicas iniciales
0.2	28/11/25	UTE ForensIA (Todos)	Revisión & Contribuciones adicionales
1.0	05/12/25	Sergio Zaera Mata	1º Borrador
1.1	16/01/26	UTE ForensIA (Todos)	Contribuciones técnicas
1.2	21/01/26	Sergio Queraltó, Jaime Castro	Revisión & Consolidación
2.0	26/01/26	Sergio Zaera Mata	2º Borrador

Cofinanciado por la Unión Europea
Fondos Europeos

ADVERTENCIA DE CONFIDENCIALIDAD Y RESPONSABILIDAD LEGAL

Este documento contiene información confidencial y secretos empresariales propiedad de la UTE FORENSIA THOT, protegidos por la Ley 1/2019 de Secretos Empresariales, el artículo 13 de la Ley de Contratos del Sector Público (LCSP) y la Directiva (UE) 2016/943 sobre protección de know-how.

Se entrega exclusivamente para la finalidad prevista en el procedimiento administrativo o contractual.

Queda terminantemente prohibida su reproducción, divulgación, cesión o uso por terceros sin autorización expresa y por escrito.

El incumplimiento de estas obligaciones puede constituir:

- Infracción contractual, con las consecuencias previstas en la LCSP.
- Responsabilidad civil y penal, conforme a la Ley 1/2019 y al Código Penal (arts. 278 y ss.).
- Acciones judiciales inmediatas, incluyendo reclamación de daños y perjuicios y medidas cautelares.

Si usted no es el destinatario autorizado, debe comunicarlo de inmediato y proceder a la eliminación del documento. Cualquier uso indebido será perseguido con el máximo rigor legal".

TABLA DE CONTENIDOS

1. INTRODUCCIÓN

1.1. Resumen ejecutivo

El presente documento F1.3.2 Documento de Arquitectura de Interoperabilidad (DAI) define la arquitectura técnica que habilita la interoperabilidad de la Plataforma THOT (Lote 1) con los sistemas del Lote 2 (Interfaces operativos y equipos para captación y tratamiento de datos en escena), con sistemas internos de la Policía Nacional y con sistemas externos nacionales e internacionales. Su objetivo es garantizar que la plataforma de inteligencia forense opera como sistema abierto, escalable y conforme a estándares, capaz de integrarse con múltiples implementaciones y evolucionar sin comprometer la estabilidad operativa.

Este documento complementa el F1.3.1 Acuerdo de Interoperabilidad, que establece el protocolo técnico común entre los Lotes 1 y 2. Mientras el F1.3.1 define *qué* se intercambia y *cómo* (formatos, endpoints, seguridad del protocolo), el F1.3.2 describe *cómo* la arquitectura interna de THOT implementa, gestiona y garantiza esa interoperabilidad a nivel de sistema, incluyendo vistas arquitectónicas (lógica, física, datos, seguridad), patrones de integración, mecanismos de sincronización offline (cuando el Lote 2 no dispone de conexión), gestión operativa de las integraciones y el plan de transición para asegurar la coexistencia con sistemas actuales.

La arquitectura de interoperabilidad de THOT se basa en los siguientes pilares técnicos:

- **Arquitectura de microservicios desacoplados** sobre Kubernetes, que permite escalar componentes de integración de forma independiente, incorporar adaptadores para nuevos sistemas externos sin afectar al núcleo y facilitar la evolución tecnológica mediante actualización modular. Los microservicios se comunican internamente mediante un bus de eventos (Apache Kafka) para garantizar resiliencia y procesamiento asíncrono.
- **API Gateway unificado** (Kong o Nginx Ingress) que centraliza la exposición de interfaces RESTful hacia el Lote 2 y otros sistemas externos, aplicando políticas transversales (autenticación, limitación de tasa, logging, versionado) de forma coherente. Este componente actúa como punto de control de acceso y auditoría para todas las integraciones entrantes.
- **Modelo de datos semántico** basado en ontologías y JSON-LD, que normaliza la información forense intercambiada con diferentes fuentes (Lote 2, bases de datos policiales, sistemas internacionales) en un formato canónico interno, preservando el significado y facilitando la interoperabilidad a nivel de contenido.
- **Capa de conectividad multiprotocolo** que soporta REST/HTTP (sincrónico), gRPC sobre HTTP/2-3 (RPC eficiente), MQTT (mensajería asíncrona para telemetría y eventos desde el campo) y WebRTC/OpenVidu (streaming de vídeo en tiempo real para asistencia remota), adaptándose a las características de cada tipo de interacción y las condiciones de red en escenarios operativos.
- **Subsistema de cadena de custodia** soportado por un registro inmutable verificable (ImmuDB), que registra de forma inalterable todos los eventos de manejo de evidencia digital, tanto los generados en los procesos de captura/ingesta como en los análisis centrales, garantizando trazabilidad forense completa, integridad criptográfica, y la admisibilidad judicial mediante firma y sellado de tiempo cualificados.

- **Mecanismos de operación offline (Lote 2)** mediante almacenamiento local seguro en dispositivos de campo, sincronización diferida al restablecer conectividad y resolución de conflictos basada en timestamps y políticas configurables (Last-Writer-Wins o revisión manual), asegurando continuidad operativa en escenarios con conectividad limitada o nula.

El diseño de esta arquitectura responde a los requisitos de interoperabilidad establecidos en el pliego (sección 4.3.2), alineándose con el marco de entendimiento (COMUN_001 a COMUN_007 del F1.3.1) y alineándose con las obligaciones de seguridad (ENS Alto), protección de datos (RGPD) y calidad (ISO 17025, ISO 17020, ISO 21043). La arquitectura soporta la integración con sistemas TIC de la Policía Nacional identificados en el pliego (PERSONAS, ABIS, EURODAC, PDyRH, entre otros), así como con sistemas forenses externos (CODIS, IBIN, bases de datos INTERPOL/EUROPOL, SIS RECAST, sistemas contemplados en Prüm II).

Criterio de verificación: La arquitectura de interoperabilidad será validada mediante pruebas de integración end-to-end que demuestren: (1) intercambio bidireccional de datos entre Lote 1 y Lote 2 según el protocolo F1.3.1; (2) consulta en tiempo real a bases de datos policiales internas; (3) sincronización offline-online con dispositivos de campo; (4) trazabilidad completa de la evidencia mediante un registro inmutable verificable (5) cumplimiento de SLAs de latencia y disponibilidad; (6) auditoría y logging de todas las transacciones de integración.

1.2. Propósito del documento

El propósito de este documento es establecer la arquitectura técnica que habilita y garantiza la interoperabilidad de la Plataforma THOT, definiendo las decisiones de diseño, componentes, flujos, interfaces y mecanismos operativos necesarios para integrar múltiples sistemas heterogéneos (Lote 2, sistemas internos de la Policía Nacional, sistemas externos nacionales e internacionales) en un ecosistema coherente, seguro, escalable y mantenible.

Este documento responde a las siguientes necesidades fundamentales:

1. **Concreción técnica de la interoperabilidad**: Traducir los requisitos de interoperabilidad del pliego (sección 4.3.2) y el protocolo acordado en el F1.3.1 en una arquitectura interna específica de THOT, detallando cómo se implementan los intercambios de datos, la seguridad, la trazabilidad y la sincronización.
2. **Visión arquitectónica multidominio**: Proporcionar vistas complementarias (lógica, física, datos, seguridad, interoperabilidad Lote 1-Lote 2) que permitan comprender el sistema desde diferentes perspectivas, facilitando la toma de decisiones técnicas, la comunicación entre equipos de desarrollo, la validación por la Policía Científica y la auditoría por entidades externas.
3. **Guía para desarrollo e integración**: Servir como especificación de referencia para los equipos de desarrollo de THOT y como base técnica para la coordinación con los adjudicatarios del Lote 2, asegurando que las implementaciones de ambos lotes converjan correctamente durante las fases de desarrollo (Fase II) y validación (Fase III).
4. **Garantía de evolución controlada**: Establecer los principios y mecanismos que permiten a THOT incorporar nuevas integraciones (nuevos sistemas del Lote 2, nuevas bases de datos policiales, nuevos servicios externos) o evolucionar tecnológicamente sin comprometer la estabilidad operativa ni la compatibilidad con integraciones existentes.

5. **Aseguramiento de cumplimiento normativo**: Demostrar cómo la arquitectura satisface los requisitos de seguridad (ENS Alto), protección de datos (RGPD), calidad (ISO 17025, ISO 17020, ISO 21043) y regulación de IA (EU AI Act) en el contexto específico de la interoperabilidad, identificando controles técnicos, puntos de auditoría y mecanismos de trazabilidad.

El documento está orientado a múltiples audiencias:

- **Equipos técnicos de desarrollo** (UTE ForensIA THOT): especificación detallada de componentes, interfaces y flujos a implementar.
- **Equipos de desarrollo del Lote 2**: definición de las interfaces que deben consumir/exponer y los mecanismos de integración a aplicar.
- **Arquitectos y responsables técnicos de la Policía Nacional**: visión de alto nivel para validar alineamiento con infraestructura existente y estrategia tecnológica corporativa.
- **Equipos de seguridad y auditoría**: controles técnicos implementados para cumplimiento normativo.
- **Dirección de proyecto y CDTI**: evidencia de avance en Fase I y base para seguimiento de Fase II.

Relación con otros entregables de Fase I:

- **F1.1.1 Prestaciones Funcionales y Técnicas**: el DAI implementa arquitectónicamente los requisitos de interoperabilidad identificados en F1.1.1 (INTEROP-1 a INTEROP-7, INT-GEN-8, INT-20A).
- **F1.2.1 Arquitectura del Sistema**: el DAI es una especialización de F1.2.1 enfocada en interoperabilidad; mientras F1.2.1 describe la arquitectura general de THOT, el DAI detalla la capa de integración.
- **F1.2.2 Modelo de Datos**: el DAI referencia el modelo de datos canónico de F1.2.2 y define los esquemas de intercambio (JSON Schema) para interoperabilidad.
- **F1.3.1 Acuerdo de Interoperabilidad**: el DAI implementa internamente el protocolo externo definido en F1.3.1.

1.3. Alcance del documento

El presente documento define la arquitectura de interoperabilidad de la Plataforma THOT (Lote 1), abarcando los siguientes aspectos técnicos y organizativos:

1.3.1. Ámbito técnico incluido

1. **Arquitectura de integración de THOT**:
    - Vista lógica: componentes de interoperabilidad (API Gateway, adaptadores, bus de eventos, servicios de sincronización).
    - Vista física: despliegue de componentes de integración en infraestructura Kubernetes.
    - Vista de datos: flujos de datos desde/hacia sistemas externos, normalización semántica, esquemas de intercambio.
    - Vista de seguridad: autenticación, autorización, cifrado, auditoría aplicada a integraciones.
    - Vista de interoperabilidad Lote 1-Lote 2: mecanismos específicos de comunicación con sistemas de campo.
2. **Interfaces y APIs**:

- Definición funcional y técnica de las APIs RESTful expuestas por THOT para Lote 2 y otros consumidores.
- Protocolos alternativos: gRPC, MQTT, WebRTC/OpenVidu (casos de uso, ventajas, implementación).
- Versionado de APIs: estrategia de evolución, compatibilidad hacia atrás, deprecación controlada.
- Documentación técnica: generación automática con OpenAPI 3.0, portal de desarrolladores.

3. **Modelos de datos compartidos**:
   - Esquemas JSON Schema para las entidades intercambiadas (asuntos, vestigios, personas, vehículos, biométricos, lugares).
   - Convenciones de nomenclatura (camelCase según F1.3.1).
   - Metadatos de trazabilidad (timestamps, identificadores únicos, firmas).
   - Mapeo entre modelo canónico interno de THOT y formatos de intercambio externos.
4. **Mecanismos de sincronización y operación offline (Lote 2)**:
   - Arquitectura de almacenamiento local en dispositivos de campo (Lote 2).
   - Colas persistentes y buffering de eventos.
   - Estrategias de sincronización diferida (detección de conflictos, resolución Last-Writer-Wins o manual).
   - Políticas de Quality of Service (QoS) aplicativo para priorización de datos.
5. **Cadena de custodia distribuida**:
   - o Integración de ImmuDB como registro inmutable.
     o Políticas de registro para el alta de eventos de custodia.
     o Microservicio CoC: interfaz REST para acceso al registro inmutable.
     o Flujo de sincronización de eventos de custodia desde Lote 2 offline
6. **Gestión operativa de integraciones**:
   - Monitorización de salud de integraciones: métricas (latencia, tasa de error, disponibilidad), herramientas (Prometheus, Grafana).
   - Gestión de incidencias: detección automática de fallos, escalamiento, registro de tickets.
   - Políticas de cambio: procedimientos para actualización de APIs, despliegue de nuevas versiones, comunicación a consumidores.
7. **Plan de transición y migración**:
   - Estrategia de coexistencia con sistemas actuales de la Policía Nacional.
   - Fases de despliegue: entorno de desarrollo, preproducción, producción; rollout por unidades territoriales.
   - Plan de reversión (rollback): procedimientos ante fallos críticos post-despliegue.
   - Impacto en procesos operativos y formación del personal.

1.3.2. Ámbito excluido (fuera de alcance)

1. **Arquitectura interna de componentes no relacionados con interoperabilidad**: el diseño detallado de servicios de IA, gestión de casos, análisis de datos que no intervienen directamente en integraciones externas se describe en F1.2.1 Arquitectura del Sistema.

2. **Especificación del protocolo de intercambio Lote 1 - Lote 2**: el protocolo técnico (endpoints, formatos, códigos de estado, seguridad de comunicación) está definido en F1.3.1 Acuerdo de Interoperabilidad; el DAI referencia ese protocolo y describe su implementación interna en THOT.
3. **Arquitectura e implementación de sistemas del Lote 2**: el DAI define las interfaces que THOT expone/consume, pero no prescribe la arquitectura interna de las soluciones del Lote 2.
4. **Integración con sistemas externos fuera del ámbito del proyecto**: el DAI se centra en integraciones con sistemas de la Policía Nacional identificados en el pliego y sistemas forenses nacionales/internacionales mencionados en requisitos (CODIS, IBIN, INTERPOL, EUROPOL, SIS RECAST, Prüm II); integraciones adicionales futuras requerirán actualización del DAI.
5. **Gestión de riesgos del proyecto**: el análisis de riesgos globales del proyecto se aborda en la memoria técnica y en F1.1.1 Prestaciones; el DAI identifica riesgos específicos de interoperabilidad y controles mitigantes, pero no sustituye el plan de riesgos global.
6. **Formación en el uso de sistemas integrados**: el impacto formativo de la interoperabilidad se describe en la sección 10.4, pero los programas formativos detallados se especifican en el entregable correspondiente (Fase I o Fase II según planificación).

1.3.3. Límites de responsabilidad y decisiones pendientes

- **Sistemas TIC de la Policía Nacional**: el DAI asume que los sistemas internos de la Policía Nacional (PERSONAS, ABIS, EURODAC, PDyRH) expondrán interfaces de consulta/actualización según estándares aplicables. Si estos sistemas requieren adaptadores legacy específicos, su desarrollo será responsabilidad conjunta THOT-Policía Nacional y se detallará durante Fase II.
- **Sistemas forenses internacionales**: el acceso a sistemas INTERPOL, EUROPOL, SIS RECAST, Prüm II está sujeto a acuerdos internacionales y procedimientos de autorización fuera del control del proyecto. El DAI define la arquitectura de conexión, pero la disponibilidad efectiva dependerá de la autorización de organismos competentes.
- **Decisiones de infraestructura física**: el DAI define requisitos lógicos de infraestructura (escalabilidad, redundancia, capacidad de red), pero la decisión sobre infraestructura on-premise vs cloud híbrido vs cloud público será tomada conjuntamente con la Policía Nacional durante el arranque de Fase II, considerando restricciones de seguridad (ENS Alto) y políticas corporativas.

Criterio de validación del alcance: Este documento se considerará completo cuando permita a un equipo técnico externo comprender: (1) qué sistemas se integran con THOT y cómo; (2) qué APIs y protocolos implementa THOT para interoperabilidad; (3) cómo se garantiza seguridad, trazabilidad y calidad en integraciones; (4) cómo se gestionan operativamente las integraciones; (5) cuál es el plan para desplegar las capacidades de interoperabilidad.

1.4. Estructura del documento

El documento se organiza en diez secciones principales que siguen una lógica de *contexto → requisitos → diseño → implementación → operación → transición*:

Sección 1 (Introducción): proporciona la visión general del documento, su propósito, alcance y esta misma guía de lectura. Establece el marco conceptual y la relación con otros entregables de Fase I.

Sección 2 (Contexto y sistemas actuales): describe el entorno tecnológico y organizativo de la Policía Nacional en el que operará THOT, identificando restricciones técnicas (infraestructura existente, tecnologías implantadas, políticas corporativas) y los principales problemas de interoperabilidad que la arquitectura debe resolver (sistemas heterogéneos, datos no estructurados, protocolos legacy, operación en campo con conectividad limitada). Esta sección justifica las decisiones arquitectónicas posteriores.

Sección 3 (Requisitos de interoperabilidad): estructura los requisitos que guían el diseño de la arquitectura de interoperabilidad, organizándolos en seis categorías: requisitos funcionales de integración (qué se integra, con quién, qué operaciones), requisitos no funcionales (rendimiento, disponibilidad, escalabilidad de integraciones), requisitos de seguridad (autenticación, autorización, cifrado, auditoría), requisitos normativos (cumplimiento ENS, RGPD, ISO), requisitos de portabilidad (independencia de infraestructura, migración futura) y requisitos de calidad de datos (validación, consistencia, trazabilidad). Cada requisito se vincula a su fuente.

Sección 4 (Principios de diseño): establece los principios arquitectónicos que gobiernan todas las decisiones de diseño de interoperabilidad: modularidad y desacoplamiento, orientación a servicios (SOA), interoperabilidad basada en estándares, seguridad por diseño, escalabilidad horizontal, resiliencia y tolerancia a fallos, observabilidad y trazabilidad. Define estándares y normas de referencia aplicables (REST/OpenAPI, JSON/JSON-LD, OAuth 2.0/mTLS, ISO 21043), criterios de extensibilidad (versionado de APIs, patrones de adaptadores) y el modelo de gobernanza interlotes (procedimientos de coordinación técnica THOT-Lote 2, gestión de cambios en interfaces compartidas).

Sección 5 (Arquitectura objetivo de interoperabilidad): presenta cinco vistas complementarias de la arquitectura:

- Vista lógica: componentes de interoperabilidad (API Gateway, adaptadores, bus de eventos, servicios de sincronización) y sus responsabilidades.
- Vista física: despliegue de componentes en Kubernetes, estrategia de alta disponibilidad.
- Vista de datos: flujos de información entre THOT y sistemas externos, normalización semántica.
- Vista de seguridad: controles de acceso, cifrado, auditoría aplicados a integraciones.
- Vista de interoperabilidad Lote 1-Lote 2: arquitectura específica de comunicación con sistemas de campo.

Cada vista incluye diagramas, descripciones de componentes, flujos y justificación de decisiones técnicas.

Sección 6 (Modelo de integración y APIs): detalla los patrones de integración aplicables (REST síncrono, MQTT asíncrono, gRPC, WebRTC), define funcionalmente las APIs clave (API de Gestión de Evidencias, API de Identificación Biométrica, Canal de Eventos y Telemetría) con ejemplos de requests/responses, especifica la estrategia de versionado de APIs (versionado en URL, compatibilidad hacia atrás, ciclo de deprecación) y describe el portal de documentación técnica para desarrolladores (generación automática con OpenAPI, sandbox de pruebas).

Sección 7 (Intercambio de datos y semántica): especifica los modelos de datos compartidos mediante JSON Schema (entidades: asunto, vestigio, persona, vehículo, lugar, biométrico), define las reglas de normalización semántica (mapeo desde formatos heterogéneos al modelo canónico THOT), describe el uso de JSON-LD para enriquecimiento semántico, detalla los mecanismos de intercambio de archivos adjuntos (imágenes, vídeos, documentos) y establece las convenciones de metadatos de trazabilidad (identificadores únicos, timestamps, checksums, firmas digitales).

Sección 8 (Gestión operativa de la interoperabilidad): describe la monitorización de integraciones (métricas clave, umbrales, dashboards, alertas), define el proceso de gestión de incidencias de integración (detección, clasificación, escalamiento, resolución, postmortem) y establece las políticas de cambio y despliegue (procedimientos para actualización de APIs, comunicación a consumidores, estrategia de rollout de nuevas versiones, validación en preproducción).

Sección 9 (Sincronización y operación en tiempo real/offline): especifica la arquitectura de soporte offline para dispositivos de campo, detallando el almacenamiento local seguro, las colas persistentes para buffering de eventos, la lógica de sincronización diferida al restablecer conectividad, las estrategias de resolución de conflictos (timestamp-based, Last-Writer-Wins, revisión manual según tipo de dato) y las políticas de Quality of Service aplicativo para priorización de sincronización de datos críticos.

Sección 10 (Plan de transición y migración): define la estrategia de coexistencia con sistemas actuales de la Policía Nacional durante el despliegue progresivo de THOT, establece las fases de despliegue de integraciones (desarrollo, preproducción, piloto, rollout territorial), especifica el plan de reversión ante fallos críticos (procedimientos de rollback, tiempos de respuesta) y evalúa el impacto en procesos operativos y formación del personal.

Audiencias objetivo por sección:

- Secciones 1-2: todas las audiencias (contexto general).
- Sección 3-4: arquitectos, responsables técnicos, auditores (requisitos y principios).
- Sección 5: arquitectos, equipos de desarrollo (visión arquitectónica).
- Sección 6-7: equipos de desarrollo THOT y Lote 2 (especificación de interfaces y datos).
- Sección 8-9: equipos de operaciones, DevOps (gestión operativa).
- Sección 10: dirección de proyecto, responsables de unidades territoriales (despliegue).

Convenciones de notación:

- Los requisitos se referencian con su ID entre corchetes, ej. [INTEROP-1], [INT-GEN-8] (del pliego).
- Los componentes arquitectónicos se escriben en **negrita** en su primera aparición, ej. **API Gateway**.
- Los nombres de tecnologías se mantienen en inglés, ej. Kubernetes, Hyperledger Fabric, Apache Kafka.
- Los ejemplos de código/JSON se presentan en bloques delimitados con triple acento grave.
- Los diagramas incluyen leyenda cuando contienen más de tres tipos de elementos.

Relación con documentación complementaria:

- Especificación OpenAPI 3.0 de las APIs THOT: se generará automáticamente desde el código y se publicará en el portal de desarrolladores durante Fase II.
- Manuales de integración para desarrolladores Lote 2: se derivarán de este DAI y del F1.3.1, incluyendo ejemplos de código, casos de uso típicos y procedimientos de troubleshooting.
- Documentación de operaciones: se derivará de las secciones 8-9 de este DAI, incluyendo runbooks, procedimientos de escalamiento y guías de resolución de incidencias.

2. CONTEXTO Y SISTEMAS ACTUALES DEL CLIENTE

Mapa de sistemas existentes; Flujos de información actuales

2.1. Restricciones técnicas y organizativas

La arquitectura de interoperabilidad de THOT debe diseñarse considerando el contexto tecnológico y organizativo real de la Policía Nacional, que impone restricciones técnicas, normativas y operativas que condicionan las decisiones de diseño. Esta sección identifica las restricciones más relevantes, justifica su impacto en la arquitectura y establece cómo THOT las aborda.

2.1.1. Restricciones de infraestructura tecnológica

Infraestructura heterogénea: La Policía Nacional opera una infraestructura TIC construida durante décadas, combinando sistemas mainframe, servidores físicos en centros de datos propios, virtualización parcial y primeros pasos hacia cloud privado. Los sistemas forenses actuales (bases de datos biométricas, sistemas de gestión de casos) están desplegados mayormente on-premise con requisitos estrictos de aislamiento de red. Esta heterogeneidad dificulta la adopción de arquitecturas nativas cloud y requiere estrategias de integración híbridas.

Impacto en THOT: La plataforma debe ser **desplegable en infraestructura on-premise** con Kubernetes, pero diseñada para portabilidad futura a cloud híbrido si la Policía Nacional evoluciona su estrategia. Los componentes de integración (API Gateway, adaptadores) deben ser capaces de comunicarse con sistemas legacy mediante protocolos estándar (REST, SOAP si es necesario) y patrones de adaptación (Enterprise Service Bus, adaptadores dedicados).

Controles aplicados:

- Arquitectura basada en contenedores Docker orquestados con Kubernetes, agnóstica de infraestructura subyacente [HW-L1-3, HW-L1-4].
- Uso de Kubernetes Federation o mecanismos equivalentes para potencial despliegue híbrido on-premise + cloud futuro.
- Definición de adaptadores específicos para integración con sistemas legacy que no soporten REST/JSON, implementados como microservicios independientes.

Ancho de banda y latencia de red: Los enlaces de red entre unidades territoriales de la Policía Científica y el centro de datos central varían en capacidad (desde 100 Mbps a 10 Gbps) y latencia (10-200 ms). Además, los dispositivos de campo (Lote 2) operan sobre redes móviles (4G/5G) con ancho de banda variable y posibles interrupciones. Esta variabilidad afecta el diseño de APIs y los mecanismos de sincronización.

Impacto en THOT: Las APIs deben optimizarse para minimizar la carga de datos (compresión, paginación eficiente, formatos binarios cuando sea apropiado). Los mecanismos de sincronización offline (Lote 2) deben ser tolerantes a desconexiones frecuentes y permitir operación autónoma en campo [INTEROP-7].

Controles aplicados:

- Uso de compresión gzip/brotli en APIs REST.
- Implementación de paginación con cursores para consultas grandes.
- Protocolo MQTT con QoS configurable para eventos desde campo (sección 6.2.3).

- Colas persistentes en dispositivos de campo para buffering de eventos durante desconexión (sección 9.1).

2.1.2. Restricciones de seguridad y cumplimiento normativo

Esquema Nacional de Seguridad (ENS) Alto: La información gestionada por THOT está clasificada como ENS Alto, lo que impone controles estrictos de acceso, cifrado, auditoría y segregación de entornos. Esto afecta la arquitectura de autenticación/autorización y los mecanismos de integración con sistemas externos.

Impacto en THOT: Todas las comunicaciones con sistemas externos deben estar cifradas (TLS 1.3 mínimo). La autenticación de sistemas integradores debe basarse en certificados digitales (mTLS) para máquina-a-máquina y OAuth 2.0 + OpenID Connect para usuarios humanos. Debe existir auditoría completa de todas las transacciones de integración [SEC-L1-3, HW-L1-8].

Controles aplicados:

- API Gateway configurado con políticas de mTLS obligatorio para APIs expuestas a Lote 2.
- Integración con sistema de gestión de identidades corporativo de la Policía Nacional (si existe) o despliegue de IdP propio (Keycloak) para autenticación federada.
- Logging centralizado de todas las peticiones API con información de identidad del solicitante, timestamp, operación, resultado (sección 8.1).
- Cifrado de datos en reposo en bases de datos y almacenamiento de objetos (AES-256).

Regulación de Protección de Datos (RGPD): El manejo de datos biométricos y filiación de personas por parte de THOT está sujeto a RGPD, especialmente artículos sobre tratamiento de categorías especiales de datos, derecho de acceso, rectificación y supresión, y obligación de Data Protection by Design.

Impacto en THOT: La arquitectura debe garantizar minimización de datos intercambiados (solo transferir información estrictamente necesaria), implementar controles de acceso basados en roles con principio de mínimo privilegio, y soportar operaciones de ciclo de vida de datos personales (retención limitada, supresión segura) [PROT-DATOS-1].

Controles aplicados:

- Políticas de retención de datos configurables por tipo de información (biométrica, filiación, evidencias) con eliminación automática al expirar plazos legales [HW-L1-9].
- Anonimización/pseudoanonimización de datos en entornos no productivos (desarrollo, formación) [HW-L1-13].
- API para ejercicio de derechos de interesados (GDPR subject rights) si aplica a sistemas integrados.

EU AI Act y Convenio Marco sobre IA: La Plataforma THOT incorpora modelos de IA de alto riesgo (reconocimiento biométrico, análisis predictivo en investigaciones penales), sujetos a obligaciones de transparencia, explicabilidad, supervisión humana y registro de operaciones.

Impacto en THOT: Los módulos de IA que intervienen en integraciones (ej. reconocimiento facial para verificación de identidad integrado con ABIS) deben registrar todas las inferencias en logs auditables, proporcionar explicaciones de decisiones (XAI) y permitir supervisión humana antes de acciones críticas [HW-L1-12, HW-L1-13].

Controles aplicados:

- Registro de cada inferencia de IA con inputs, outputs, modelo utilizado, timestamp, usuario solicitante (si aplica) en base de datos auditable.
- Implementación de técnicas XAI (SHAP/LIME) en APIs de IA que interactúan con sistemas externos.
- Flujos de trabajo que requieren aprobación humana antes de difundir resultados de IA a sistemas externos o generar alertas automáticas en integraciones.

2.1.3. Restricciones organizativas y operativas

Distribución territorial: La Policía Científica opera en más de 100 ubicaciones (brigadas provinciales, unidades centrales) con niveles heterogéneos de capacidad técnica local. El despliegue de capacidades de interoperabilidad debe ser escalable territorialmente y operable con personal de diferentes niveles de especialización técnica.

Impacto en THOT: La arquitectura debe soportar despliegue federado (cada ubicación puede tener componentes locales de caché/sincronización si es necesario) y las interfaces de usuario relacionadas con interoperabilidad deben ser intuitivas, minimizando la necesidad de formación técnica profunda para uso operativo [UX-3, UX-4].

Controles aplicados:

- Despliegue centralizado con acceso remoto para la mayoría de unidades (modelo hub-and-spoke).
- Posibilidad de desplegar nodos edge de THOT en ubicaciones críticas (p.ej. laboratorios centrales) con sincronización al hub central (a decidir durante Fase II según análisis de carga).
- Interfaz web responsive y aplicaciones móviles (Lote 2) con diseño centrado en usuario, guías contextuales y asistentes virtuales [INT-42].

Continuidad operativa 24/7: La Policía Científica opera continuamente sin ventanas de mantenimiento programadas en producción (salvo emergencias). Las actualizaciones de THOT, incluyendo cambios en APIs de integración, deben realizarse sin interrumpir el servicio.

Impacto en THOT: La arquitectura debe soportar despliegues blue-green o canary, permitiendo actualizar componentes de integración mientras versiones anteriores siguen operativas. El versionado de APIs debe garantizar compatibilidad hacia atrás durante periodo de transición (sección 6.3).

Controles aplicados:

- Kubernetes con múltiples réplicas de cada microservicio de integración, permitiendo rolling updates sin downtime.
- API Gateway configurado para enrutar peticiones a múltiples versiones de API simultáneamente (ej. /v1/ y /v2/ coexisten).
- Procedimientos de despliegue con validación en preproducción, rollout gradual y rollback automático ante incremento de tasa de error (sección 10.2).

2.1.4. Restricciones de interoperabilidad con sistemas internos

Sistemas críticos de la Policía Nacional: THOT debe integrarse con múltiples sistemas TIC corporativos, incluyendo (lista no exhaustiva basada en el pliego):

- **PERSONAS**: base de datos nacional de filiación e identidad.

Cofinanciado por la Unión Europea
Fondos Europeos

- **ABIS (Automated Biometric Identification System)**: sistema de identificación por huellas dactilares, registro fotográfico.
- **EURODAC**: sistema europeo de huellas dactilares para solicitantes de protección internacional.
- **PDyRH (Policía de Fronteras y Derechos Humanos)**: gestión de migraciones y fronteras.
- **Sistemas de gestión de casos policiales**: aplicaciones de investigación criminal.

Estos sistemas tienen niveles heterogéneos de modernidad tecnológica (algunos ofrecen APIs REST modernas, otros solo SOAP o acceso directo a BBDD, algunos requieren VPN dedicadas) y diferentes responsables de mantenimiento (algunas unidades de la Policía Nacional, otras externas).

Impacto en THOT: La arquitectura debe ser flexible para adaptarse a diferentes tecnologías de integración y no asumir que todos los sistemas ofrecen APIs REST/JSON. Se requiere una capa de adaptadores que normalice la heterogeneidad [INTEROP-3].

Controles aplicados:

- Definición de **Servicio de Espacio de Datos** (según memoria técnica) con adaptadores específicos para cada sistema interno, implementados como microservicios independientes que traducen protocolos/formatos heterogéneos a modelo canónico THOT.
- Uso de **Enterprise Service Bus (ESB)** ligero o capacidades de **API Gateway** (Kong con plugins de transformación) para mediación entre protocolos cuando no justifique microservicio dedicado.
- Documentación de cada adaptador con requisitos de conectividad, formato de datos, autenticación, limitaciones conocidas.

Procedimientos de autorización para acceso a sistemas internos: El acceso de THOT a sistemas corporativos sensibles (ej. PERSONAS, ABIS) requiere autorizaciones formales y configuración de credenciales/certificados por parte de administradores de esos sistemas. Esto implica coordinación con múltiples unidades y tiempos de configuración no controlables por el proyecto THOT.

Impacto en THOT: El diseño de integraciones con sistemas internos debe considerar que algunas no estarán disponibles en Fase II temprana (desarrollo) y se habilitarán progresivamente. La arquitectura debe permitir "stubs" o simuladores de estos sistemas para desarrollo/pruebas [RVE-3a].

Controles aplicados:

- Implementación de simuladores (mocks) de sistemas externos para entornos de desarrollo y preproducción, con comportamiento configurable.
- Plan de integración por fases: (1) mocks en desarrollo, (2) integración real en preproducción con subconjunto de datos, (3) integración completa en producción.
- Coordinación con responsables de sistemas TIC corporativos mediante Comité Técnico de Integración (a establecer en Fase II).

2.1.5. Restricciones de interoperabilidad internacional

Sistemas forenses internacionales: El pliego menciona la necesidad de interoperabilidad con sistemas INTERPOL, EUROPOL, SIS RECAST (Sistema de Información Schengen), Prüm II y sistemas forenses específicos (CODIS para ADN, IBIN para balística). El acceso a estos sistemas está regulado por acuerdos internacionales y procedimientos de autorización que no están bajo control del proyecto THOT.

Impacto en THOT: La arquitectura debe diseñarse para *potencial* integración con estos sistemas, definiendo interfaces y adaptadores genéricos, pero aceptando que la disponibilidad efectiva dependerá de autorizaciones políticas externas al proyecto [INT-20A].

Controles aplicados:

- Definición de interfaces abstractas para integración con sistemas forenses (ej. ForensicDatabaseAdapter) que puedan implementarse para CODIS, IBIN, INTERPOL, etc.
- Priorización de integración con sistemas nacionales en Fase II, dejando sistemas internacionales para fases posteriores o como extensiones futuras según disponibilidad de acuerdos.
- Documentación de requisitos técnicos de integración con sistemas internacionales basada en estándares públicos disponibles (ej. especificaciones INTERPOL I-24/7).

2.1.6. Resumen de restricciones y decisiones arquitectónicas derivadas

La siguiente tabla resume las restricciones identificadas y su traducción en decisiones arquitectónicas vinculantes para THOT:

Restricción	Origen	Decisión arquitectónica	Sección del DAI
Infraestructura heterogénea on-premise	ENS Alto, infraestructura actual	Arquitectura basada en contenedores/Kubernetes, portable, adaptadores para sistemas legacy	5.1, 5.2
Ancho de banda variable y operación offline (Lote 2)	Despliegue territorial, dispositivos campo	Compresión de datos, colas persistentes, sincronización diferida	6.2, 9.1, 9.2
ENS Alto (seguridad máxima)	Normativa	mTLS, OAuth 2.0, cifrado E2E, auditoría completa	5.4, 6.2, 8.1
RGPD (protección de datos)	Normativa	Minimización de datos, retención limitada, trazabilidad	3.3, 7.1
EU AI Act (IA regulada)	Normativa	XAI, registro de inferencias, supervisión humana	3.4, 5.1
Distribución territorial 100+ ubicaciones	Organización	Despliegue federado centralizado, UI intuitiva	5.2, 10.2
Operación 24/7 sin downtime	Operativa	Despliegues blue-green, versionado de APIs compatible	6.3, 8.3, 10.3
Heterogeneidad de sistemas TIC internos	Legado tecnológico	Capa de adaptadores, ESB/mediación	5.1, 6.1

Cofinanciado por la Unión Europea
Fondos Europeos

Restricción	Origen	Decisión arquitectónica	Sección del DAI
Autorizaciones para sistemas corporativos	Gobernanza	Integración por fases, uso de mocks	10.2
Autorizaciones para sistemas internacionales	Acuerdos internacionales	Interfaces abstractas, integración futura	6.1, 10.2

Tabla 1. Resumen de restricciones (técnicas, normativas y operativas) y su traducción en decisiones arquitectónicas vinculantes para THOT. *Traza cada restricción a su origen y a la decisión correspondiente. Incluye la referencia a las secciones del DAI donde se detalla cada decisión para facilitar verificación, auditoría y seguimiento de cumplimiento.*

Estas decisiones arquitectónicas fundamentales se desarrollan en detalle en las secciones siguientes del DAI.

2.2. Principales problemas de interoperabilidad detectados

Esta sección identifica los problemas de interoperabilidad más relevantes que existen en el ecosistema actual de sistemas forenses de la Policía Nacional, los cuales justifican el diseño de la arquitectura de THOT y establecen los objetivos que debe cumplir la plataforma para mejorar sustancialmente la situación actual.

2.2.1. Fragmentación de sistemas y silos de información

Descripción del problema: Actualmente, diferentes unidades de la Policía Científica (inspección técnico-policial, identificación biométrica, laboratorios especializados) utilizan sistemas de información independientes, muchos de ellos no interconectados o con integración manual. Esto resulta en:

- **Duplicación de datos**: la misma información (ej. identificación de una persona, vestigios relacionados con un caso) debe introducirse múltiples veces en sistemas diferentes.
- **Falta de visión integrada**: los analistas no pueden acceder desde un único punto a toda la información forense relevante de un caso, debiendo consultar múltiples aplicaciones de forma secuencial.
- **Dificultad para correlación de datos**: identificar relaciones entre casos, personas o vestigios requiere esfuerzo manual de cruces de información entre sistemas.

Evidencia de origen: Esta problemática se identifica implícitamente en el pliego (sección 2.1.1 "Plataforma interoperable de servicios de inteligencia forense") cuando se demanda "integrar, analizar y explotar la información procedente de diferentes escenas en tiempo real, capaz de asociar datos, crear redes de conexión e identificar patrones" y "intercambio de información rápido, tanto interna como externamente".

Impacto operativo: Incrementa tiempos de respuesta en investigaciones, genera inconsistencias en datos, dificulta la generación de inteligencia forense (detección de patrones, series delictivas) y aumenta la carga administrativa del personal.

Solución THOT: La plataforma actúa como **sistema unificador de inteligencia forense** implementando:

- **Servicio de Espacio de Datos** centralizado que integra información de múltiples fuentes mediante adaptadores [INT-GEN-2, INT-GEN-4].
- **Registro único de asuntos** que vincula toda la información relacionada (vestigios, análisis, personas, vehículos, lugares) desde cualquier fuente, evitando duplicación.
- **Modelo de datos semántico** que normaliza información heterogénea en estructura canónica, permitiendo consultas unificadas [INTEROP-4].
- **Interfaz de usuario única** que proporciona acceso integrado a toda la información forense del caso sin necesidad de cambiar entre aplicaciones [INT-41].

Criterio de verificación: En Fase III, una consulta desde la UI de THOT debe retornar información agregada de al menos tres fuentes heterogéneas (ej. datos de ITP desde Lote 2, identificación biométrica desde ABIS, filiación desde PERSONAS) en un tiempo de respuesta <3 segundos, demostrando integración efectiva.

2.2.2. Intercambio de información no estructurado y manual

Descripción del problema: El intercambio de resultados de análisis forenses entre unidades territoriales y con otros cuerpos policiales o judiciales se realiza frecuentemente mediante documentos PDF, correos electrónicos o llamadas telefónicas. Este proceso:

- **No es trazable**: difícil saber quién accedió a qué información, cuándo y con qué finalidad.
- **No permite explotación automatizada**: la información encapsulada en PDF no es procesable por sistemas de IA/análisis.
- **Genera retrasos**: el ciclo de solicitud-envío-recepción-procesamiento es lento.

Evidencia de origen: El pliego (4.3.2.3 requisitos de interoperabilidad entre lotes) exige "digitalización y automatización de los procesos de intercambio de información científico-forense... siguiendo estándares y protocolos establecidos" [INTEROP-1]. La memoria técnica (sección 2.1.2.7 Servicio de Comunicaciones) propone "comunicación bidireccional, segura y resiliente".

Impacto operativo: Limita la velocidad de investigaciones, dificulta el cumplimiento de plazos legales (ej. retención de sospechosos), impide análisis cruzados automatizados (ej. detección de modus operandi similares) y genera fricción en colaboración con INTERPOL/EUROPOL.

Solución THOT: Implementación de **intercambio automatizado estructurado** mediante:

- **APIs REST para intercambio de resultados forenses** con formatos estandarizados (JSON Schema) [INTEROP-1, sección 6.2 de este DAI].
- **Suscripción a eventos** mediante MQTT: sistemas externos pueden suscribirse a notificaciones automáticas cuando se genere un resultado relevante (ej. match biométrico) [sección 6.2.3].
- **Portal web para organismos externos** (judiciales, otros cuerpos) con autenticación federada y acceso controlado a informes.
- **Auditoría completa de intercambios**: registro inmutable de qué información se compartió, con quién, cuándo, bajo qué autorización legal [HW-L1-11].

Criterio de verificación: En Fase III, simular envío de resultado de análisis balístico desde THOT a sistema INTERPOL mock vía API REST en <5 segundos, con registro auditable completo de la transacción.

2.2.3. Operación en campo con conectividad limitada

Descripción del problema: Los equipos de la Policía Científica desplegados en escenas del delito (especialmente en zonas rurales, sótanos, zonas de catástrofe) enfrentan frecuentemente conectividad de red limitada o nula. Con sistemas actuales que requieren conectividad permanente:

- **No pueden registrar información en tiempo real**: deben tomar notas en papel y transcribirlas posteriormente en sistemas centrales, generando retraso y riesgo de pérdida de información.
- **No pueden consultar bases de datos en campo**: imposible verificar identidad de sospechosos o consultar antecedentes en tiempo real si no hay conexión.
- **No pueden recibir asistencia remota**: los expertos en la central no pueden guiar al personal en campo mediante visualización de la escena en tiempo real.

Evidencia de origen: El pliego (sección 3.1.1 Interfaces operativos) exige "permitir asistencia a la escena en remoto en contextos de crisis y modo degradado de trabajo" y "trabajo en modo degradado como trabajo en condiciones limitadas... falta o sin posibilidad de comunicación". El F1.3.1 (sección 4.5 Comportamiento offline) define: "Cuando un sistema (campo) no tiene conexión, debe almacenar localmente toda la información generada hasta que pueda sincronizar".

Impacto operativo: Reduce la eficacia de intervenciones en campo, incrementa riesgo de pérdida de evidencias, impide asistencia remota por expertos, genera retraso en inicio de análisis (información no llega a central hasta retorno de equipo de campo).

Solución THOT: Implementación de **arquitectura offline-first para Lote 2** con sincronización inteligente:

- **Almacenamiento local cifrado** en dispositivos de campo (tablets, smartphones) con subconjunto de información crítica (bases de datos biométricas reducidas, protocolos de actuación) [INTEROP-7].
- **Colas persistentes** para buffering de eventos generados en campo (registros de vestigios, fotografías, vídeos, cadena de custodia) durante desconexión [sección 9.1].
- **Sincronización automática diferida** al restablecer conectividad, con priorización de datos críticos (QoS aplicativo) [sección 9.2].
- **Resolución de conflictos** basada en timestamps y políticas configurables para casos donde información se modifica en campo y en central simultáneamente [sección 9.1].

Criterio de verificación: En Fase III, realizar demostración de captura de 10 vestigios en dispositivo de campo sin conexión, simular desplazamiento y reconexión, verificar sincronización automática completa en <2 minutos con integridad verificada (checksums coincidentes).

2.2.4. Falta de trazabilidad y cadena de custodia digital

Descripción del problema: La cadena de custodia de evidencias físicas está bien establecida (etiquetado, registros manuales, firmas), pero la cadena de custodia de **evidencia digital** (ficheros capturados en campo, resultados de análisis, informes) es menos robusta:

- **No hay registro inmutable** de quién accedió/modificó/copió un fichero digital.
- **Dificultad para demostrar integridad** de evidencia digital ante tribunal: ¿cómo probar que una imagen no fue alterada entre captura y presentación judicial?
- **No hay interoperabilidad** entre sistemas de custodia de diferentes unidades o países.

Cofinanciado por la Unión Europea · Fondos Europeos

Evidencia de origen: El pliego (4.1.1.2 ITP) exige "consolidar y automatizar los procesos de etiquetado de vestigios y cadena de custodia siguiendo estándares internacionales... sistema de gestión de calidad de Policía Científica" [INS-EJE-6].

Impacto operativo: Riesgo de invalidación de evidencia digital por falta de trazabilidad demostrable, dificultad para auditorías (internas, judiciales), fricción en colaboración internacional (ej. EUROPOL no acepta evidencia si no cumple estándares de custodia).

Solución THOT: Implementación de cadena de custodia inmutable mediante registro inmutable (ImmuDB):

- **ImmuDB** como registro inmutable que registra todos los eventos de custodia [sección 5.1]
- **Validaciones** que verifican que cada evento de custodia cumple reglas predefinidas (ej. solo personal autorizado puede modificar estado de vestigio).
- **Microservicio Ledger CoC** con API REST que permite a Lote 2 y otros sistemas registrar eventos de custodia y consultar historial completo [INT-GEN-7B].
- **Integración con estándar ISO 21043** (investigación forense – principios y requisitos) para asegurar aceptación internacional [sección 4.2].

Criterio de verificación: En Fase III, registrar 100 eventos de custodia (captura, transferencia, análisis, emisión de informe) de un vestigio digital, consultar historial completo desde el registro inmutable, exportar cadena de custodia certificada (PDF firmado digitalmente) admisible ante tribunal.

2.2.5. Dificultad para integración con sistemas internacionales

Descripción del problema: Aunque España participa en sistemas de intercambio de información policial europeos (SIS, EUROPOL) e internacionales (INTERPOL), la integración técnica de sistemas nacionales con estos sistemas es compleja:

- **Protocolos heterogéneos**: cada sistema internacional usa formato/protocolo propio (ej. mensajes XML específicos de INTERPOL, formatos SIENA de EUROPOL).
- **Procedimientos manuales**: en muchos casos, el intercambio requiere acceso manual a portal web del sistema internacional, copiar-pegar información.
- **Latencia alta**: el ciclo de consulta a sistema internacional puede tardar horas o días.

Evidencia de origen: El pliego (4.1.2 Requisitos de interoperabilidad policial) menciona "interacción con sistemas y bases de datos como las contempladas en el Reglamento PrümII, así como las de INTERPOL, EUROPOL, Sistema de Información Schengen (SIS RECAST)" [INTEROP-2].

Impacto operativo: Limita la capacidad de detectar conexiones internacionales de casos (ej. sospechoso con antecedentes en otro país), genera fricción en investigaciones transnacionales, dificulta cumplimiento de obligaciones de intercambio de información según Tratados europeos.

Solución THOT: Implementación de **adaptadores para sistemas internacionales** con normalización de protocolos:

- **Servicio de Espacio de Datos** incluye adaptadores específicos para INTERPOL I-24/7, EUROPOL SIENA, SIS, Prüm II (a desarrollar progresivamente según disponibilidad de acuerdos y acceso técnico) [INT-GEN-8, INT-20A].
- **Modelo de datos ontológico** que mapea información forense española a formatos requeridos por sistemas internacionales y viceversa [sección 7.1].

- **Consultas asíncronas** con seguimiento de estado: THOT envía consulta a sistema internacional, registra solicitud, notifica al usuario cuando llega respuesta (puede tardar horas), evitando bloqueo de flujo de trabajo [sección 6.1].

Criterio de verificación: Durante Fase II-III, implementar adaptador para al menos un sistema internacional (a determinar según disponibilidad, prioridad: INTERPOL), demostrar envío de consulta biométrica y recepción de respuesta, con mapeo correcto de formatos.

2.2.6. *Falta de capacidades de tiempo real en análisis forense*

Descripción del problema: El modelo tradicional de trabajo forense es secuencial y por lotes: (1) captura de vestigios en escena, (2) transporte físico a laboratorio, (3) análisis en laboratorio, (4) emisión de informe, (5) difusión de resultado. Este ciclo puede tardar días o semanas. En escenarios críticos (terrorismo, secuestros, catástrofes), este retraso es inaceptable.

Evidencia de origen: El pliego (2.1.1 Plataforma interoperable) exige "integrar, analizar y explotar la información procedente de diferentes escenas en **tiempo real**" y "generación de inteligencia en el mismo escenario del delito". La memoria técnica (2.1.2.5 Servicio de Alertas) propone "sistema de alertas avanzado y proactivo".

Impacto operativo: Reduce la efectividad de intervenciones críticas (no se puede actuar sobre inteligencia forense hasta que llegue informe formal), impide "intelligence-led policing" (toma de decisiones operativas basadas en inteligencia forense temprana), limita capacidad de respuesta ante amenazas graves.

Solución THOT: Implementación de **flujos de trabajo en tiempo real** con alertas proactivas:

- **Procesamiento de eventos en streaming** mediante Apache Kafka: datos capturados en campo (Lote 2) se envían a THOT mediante MQTT/REST, se ingestan en Kafka, se procesan en tiempo real por microservicios de análisis [sección 5.1].
- **Análisis automático con IA** en datos entrantes: ej. reconocimiento facial de sospechoso capturado en campo se procesa inmediatamente contra ABIS, resultado (match/no match) se genera en <10 segundos [INT-20, INT-22].
- **Sistema de alertas** que notifica automáticamente a investigadores cuando se detecta patrón relevante (ej. match biométrico, detección de material peligroso, correlación con caso anterior) [INT-17, INT-18].
- **Dashboard de situación en tiempo real** para responsables de operación, mostrando estado de escenas activas, vestigios capturados, análisis en curso, alertas generadas [INT-41].

Criterio de verificación: En Fase III, capturar imagen facial de persona en dispositivo de campo (Lote 2), enviar a THOT, procesar reconocimiento facial contra base de datos simulada (10.000 registros), generar alerta si hay match, todo el ciclo end-to-end en <30 segundos.

2.2.7. *Resumen de problemas y mapeo a soluciones THOT*

Problema detectado	Impacto operativo	Solución THOT	Req. relacionado	Sección DAI
Fragmentación de sistemas, silos de información	Duplicación datos, falta de visión	Servicio Espacio de Datos, registro único asuntos, modelo semántico	INT-GEN-2, INT-GEN-4, INTEROP-4	5.1, 5.3, 7.1

Cofinanciado por la Unión Europea Fondos Europeos

Problema detectado	Impacto operativo	Solución THOT	Req. relacionado	Sección DAI
	integrada, dificultad correlación			
Intercambio no estructurado y manual	No trazable, no explotable por IA, retrasos	APIs REST, suscripción eventos MQTT, portal web, auditoría	INTEROP-1, HW-L1-11	6.2, 8.1
Operación en campo con conectividad limitada	No registro en tiempo real, no consultas en campo, no asistencia remota	Almacenamiento local, colas persistentes, sincronización diferida	INTEROP-7	9.1, 9.2
Falta de trazabilidad y CoC digital	Riesgo invalidación evidencia, dificultad auditorías, fricción colaboración internacional	(ImmuDB), validaciones de registro, ISO 21043	INS-EJE-6, INT-GEN-7B	5.1, 5.4
Dificultad integración con sistemas internacionales	Limita detección conexiones internacionales, fricción investigaciones transnacionales	Adaptadores INTERPOL/EUROPOL, modelo ontológico, consultas asíncronas	INT-GEN-8, INT-20A, INTEROP-2	5.1, 6.1, 7.1
Falta de capacidades tiempo real	Reduce efectividad intervenciones críticas, impide intelligence-led policing	Procesamiento eventos streaming (Kafka), análisis automático IA, sistema alertas	INT-20, INT-22, INT-17	5.1, 6.2.3

Tabla 2. Resumen de problemas de interoperabilidad identificados, su impacto operativo y el mapeo a soluciones específicas de THOT. *Relaciona cada problema con la capacidad o componente que lo resuelve y traza la vinculación con los requisitos aplicables.*

Estos problemas de interoperabilidad identificados, junto con las restricciones técnicas y organizativas de la sección 2.1, justifican y guían el diseño de la arquitectura de interoperabilidad de THOT que se detalla en las secciones siguientes.

Cofinanciado por la Unión Europea
Fondos Europeos

3. REQUISITOS DE INTEROPERABILIDAD

Esta sección estructura los requisitos que gobiernan el diseño de la arquitectura de interoperabilidad de THOT, organizándolos en seis categorías para facilitar su trazabilidad y verificación. Cada requisito se vincula a su fuente canónica (pliego, memoria técnica, F1.3.1, registro de requisitos) y se especifica su criterio de aceptación.

Los requisitos aquí presentados complementan y concretan los requisitos generales del sistema (F1.1.1 Prestaciones Funcionales y Técnicas) en el ámbito específico de la interoperabilidad.

3.1. Requisitos funcionales de integración

(qué información se intercambia, con quién y para qué: descarga y visualización de atestados policiales, consulta de datos de personas y documentos, registro oficial de documentos vía telemática etc.)

Estos requisitos definen *qué* debe integrarse, *con quién* y *qué operaciones* deben soportarse.

3.1.1. RF-INTEROP-01: Integración con sistemas del Lote 2

Descripción: THOT debe integrarse bidireccionalmente con los sistemas del Lote 2 (interfaces operativos y equipos de captación de datos en escena) según el protocolo definido en F1.3.1, soportando:

- Recepción de datos capturados en campo (vestigios, fotografías, vídeos, lecturas de sensores, eventos de cadena de custodia) vía REST, MQTT o gRPC.
- Envío de resultados de análisis, alertas y consultas desde THOT hacia dispositivos de campo.
- Streaming de vídeo bidireccional (WebRTC) para asistencia remota.
- Operación offline (Lote 2) con sincronización diferida.

Fuente: Pliego sección 4.3.2.3, F1.3.1 sección 6, memoria técnica 2.1.2.7.

ID requisitos relacionados: INTEROP-3, INTEROP-6, INTEROP-7, INS-EJE-2.

Criterio de aceptación:

- Caso de prueba 1: Dispositivo Lote 2 envía JSON con vestigio a endpoint REST THOT, recibe respuesta HTTP 201 con ID de vestigio asignado, vestigio queda registrado en BBDD THOT con trazabilidad completa (timestamp, usuario, dispositivo origen).
- Caso de prueba 2: THOT envía alerta a dispositivo Lote 2 conectado vía MQTT, dispositivo recibe mensaje en <5 segundos, se registra confirmación de entrega.
- Caso de prueba 3: Establecer sesión WebRTC entre UI THOT (experto en central) y app Lote 2 (agente en campo), transmitir vídeo bidireccionalmente con latencia <500 ms en red 4G simulada.
- Caso de prueba 4: Dispositivo Lote 2 genera 50 eventos sin conexión, reconecta, sincroniza automáticamente, THOT los recibe en orden correcto (según timestamp) sin pérdida.

Prioridad: 5 (requisito explícito pliego).

3.1.2. RF-INTEROP-02: Integración con sistemas TIC de Policía Nacional

Descripción: THOT debe integrarse con los siguientes sistemas corporativos de la Policía Nacional (lista mínima):

- **PERSONAS**: base de datos de filiación, consulta por NIF/NIE/pasaporte, recepción de datos de nuevas identificaciones.
- **ABIS**: sistema biométrico dactiloscópico y facial, envío de solicitudes de cotejo 1:N, recepción de resultados (match/no match + score).
- **EURODAC**: sistema europeo de huellas dactilares, consulta según protocolo Prüm II.
- **PDyRH**: gestión de migraciones, consulta de movimientos fronterizos.
- **Sistemas de gestión de casos policiales**: sincronización de información de asuntos (pendiente de identificar sistemas específicos en Fase II).

La integración debe:

- Consultar estos sistemas en tiempo real desde THOT (latencia <3 segundos para consultas simples).
- Enviar actualizaciones (cuando aplique y esté autorizado) desde THOT a estos sistemas.
- Auditar todas las transacciones.

Fuente: Pliego sección 4.1.2 INS-EJE-4, RES-2, RES-4.

ID requisitos relacionados: INTEROP-1, INTEROP-2, INT-GEN-8.

Criterio de aceptación:

- Caso de prueba 1: Desde UI THOT, realizar consulta a sistema PERSONAS simulado por NIF, recibir respuesta con datos de filiación en <3 segundos, mostrar resultado en pantalla, registrar consulta en log de auditoría.
- Caso de prueba 2: Enviar huella dactilar desde THOT a ABIS simulado para cotejo 1:N contra 100.000 registros, recibir resultado (lista de candidatos con scores) en <10 segundos.
- Caso de prueba 3: Intentar consulta a PERSONAS sin credenciales válidas, THOT rechaza operación y registra intento no autorizado en log de seguridad.

Prioridad: 5 (requisito explícito pliego).

3.1.3. RF-INTEROP-03: Integración con sistemas forenses internacionales

Descripción: THOT debe diseñarse para potencial integración con sistemas forenses nacionales e internacionales:

- **CODIS** (Combined DNA Index System): intercambio de perfiles genéticos.
- **IBIN** (Interpol Ballistics Information Network): intercambio de información balística.
- **INTERPOL I-24/7**: mensajería policial internacional.
- **EUROPOL SIENA**: intercambio seguro de información criminal.

- **SIS RECAST**: Sistema de Información Schengen.
- **Prüm II**: intercambio automatizado de perfiles biométricos entre Estados UE.

La arquitectura debe incluir **interfaces abstractas** (patrones de adaptadores) que permitan implementar conectores específicos para cada sistema cuando se disponga de autorización y acceso técnico, sin requerir rediseño de THOT.

Fuente: Pliego sección 4.1.2 INT-20A, INT-GEN-8.

ID requisitos relacionados: INTEROP-2.

Criterio de aceptación:

- Demostrar en Fase II que se puede implementar un adaptador para un sistema internacional simulado (ej. mock de INTERPOL) en <2 semanas de desarrollo, integrándolo con THOT sin modificar núcleo de la plataforma, solo añadiendo nuevo microservicio adaptador.
- Documentar especificación de interfaz abstracta ForensicDatabaseAdapter con métodos: query(), submit(), getStatus(), implementables para cualquier sistema externo.

Prioridad: 4 (requisito identificado por consorcio, no explícito en pliego pero derivado de INTEROP-2).

3.1.4. RF-INTEROP-04: Digitalización de ciclo completo de evidencias

Descripción: THOT debe digitalizar y automatizar el ciclo completo de procesamiento de evidencias:

1. **Entrada**: captura de vestigio en campo (Lote 2) → registro automático en THOT con metadatos (ubicación GPS, timestamp, fotógrafo, etc.).
2. **Etiquetado**: asignación automática de ID único, impresión de etiqueta física con QR/RFID vinculado a registro digital.
3. **Cadena de custodia**: registro inmutable en ImmuDB de cada transferencia/acceso al vestigio.
4. **Análisis**: asignación a experto, registro de método aplicado, resultados estructurados (no solo PDF).
5. **Difusión**: envío automático de resultado a destinatarios autorizados (investigadores, jueces) vía API o portal.
6. **Retención/destrucción**: gestión automatizada de plazos legales, eliminación segura certificada.

Fuente: Pliego 4.3.2.3 INTEROP-1, INS-EJE-6, HW-L1-9.

ID requisitos relacionados: INT-GEN-7A, INT-GEN-7B, INT-GEN-7C.

Criterio de aceptación:

- Caso de prueba end-to-end: simular ciclo completo (captura → análisis → difusión) de un vestigio, verificar que cada paso se registra automáticamente, cadena de custodia completa consultable, resultado accesible vía API para sistema externo autorizado.

Prioridad: 5 (requisito explícito pliego).

3.1.5. RF-INTEROP-05: Consulta en tiempo real desde campo

Descripción: Personal de la Policía Científica desplegado en campo (usando dispositivos Lote 2) debe poder consultar en tiempo real:

- Bases de datos biométricas (ABIS, EURODAC): envío de huella/foto desde campo, recepción de resultado match/no match en <30 segundos.
- Bases de datos de filiación (PERSONAS): consulta por documento de identidad.
- Información de casos: consultar detalles de asunto en curso (vestigios ya registrados, análisis previos).

La consulta debe funcionar con conectividad 4G/5G (latencia 50-200 ms) y optimizarse para minimizar consumo de datos móviles.

Fuente: Pliego 3.1.1 Interfaces operativos INS-EJE-4.

ID requisitos relacionados: INTEROP-6.

Criterio de aceptación:

- Desde dispositivo Lote 2 con conexión 4G simulada (100 ms latencia, 10 Mbps ancho de banda), capturar huella dactilar, enviar a THOT, cotejar contra ABIS simulado (10.000 registros), recibir resultado en <30 segundos, mostrar en pantalla dispositivo.

Prioridad: 5 (requisito explícito pliego).

3.2. Requisitos no funcionales

(rendimiento, disponibilidad, latencia, volúmenes de datos, compatibilidad con protocolos legacy y nuevos; respeto de plazos de respuesta según caso: desde la inmediatez 24/7, hasta los 7 días)
Estos requisitos definen características de calidad de las integraciones (rendimiento, disponibilidad, escalabilidad).

3.2.1. RNF-INTEROP-01: Latencia de APIs REST

Descripción: Las APIs REST expuestas por THOT para Lote 2 y otros consumidores deben cumplir los siguientes SLAs de latencia (percentil 95):

- Operaciones de escritura sencillas (POST vestigio, registro evento CoC): <500 ms.
- Operaciones de lectura sencillas (GET vestigio por ID, consulta lista de vestigios de un asunto): <300 ms.
- Operaciones de búsqueda complejas (búsqueda full-text en todos los vestigios): <2 segundos.
- Operaciones de análisis con IA (reconocimiento facial, análisis de patrones): <10 segundos para datasets pequeños (<100 elementos), <60 segundos para datasets medianos (<1.000 elementos).

La latencia se mide desde recepción de petición HTTP en API Gateway hasta envío de respuesta HTTP completa.

Fuente: Pliego 4.3.2.3 "transmisión en tiempo real" UX-5.

ID requisitos relacionados: INTEROP-6, RVE-2b.

Criterio de aceptación:

- Pruebas de rendimiento en entorno de preproducción: ejecutar 1.000 peticiones POST vestigio concurrentes, medir latencia percentil 95, verificar <500 ms.
- Ejecutar 500 peticiones GET vestigio concurrentes, verificar percentil 95 <300 ms.

Prioridad: 5 (requisito explícito pliego).

3.2.2. RNF-INTEROP-02: Disponibilidad de servicios de integración

Descripción: Los servicios de interoperabilidad de THOT (API Gateway, microservicios de adaptadores, bus de eventos Kafka) deben garantizar disponibilidad ≥99.5% mensual (downtime máximo permitido ~3.6 horas/mes).

La medición excluye ventanas de mantenimiento programadas (máximo 4 horas/mes, notificadas con 7 días de antelación) y caídas debidas a factores externos (fallo de infraestructura de red de Policía Nacional, caída de sistemas externos integrados).

Fuente: Pliego HW-L1-3, memoria técnica 2.1.2.8 alta disponibilidad.

ID requisitos relacionados: HW-L1-7.

Criterio de aceptación:

- Durante periodo de observación de 1 mes en producción (Fase III), monitorizar uptime de API Gateway y servicios críticos, calcular disponibilidad = (tiempo total - downtime no planificado) / tiempo total, verificar ≥99.5%.

Prioridad: 5 (requisito explícito pliego).

3.2.3. RNF-INTEROP-03: Escalabilidad horizontal de integraciones

Descripción: La arquitectura de integración debe escalar horizontalmente para soportar incrementos de carga sin degradación de rendimiento:

- **Carga base** (operación normal): 100 dispositivos Lote 2 concurrentes, 50 peticiones API/segundo, 500 mensajes MQTT/segundo.
- **Carga pico** (operación en emergencia, ej. atentado): 500 dispositivos concurrentes, 200 peticiones API/segundo, 2.000 mensajes MQTT/segundo.

El escalado debe ser automático (Horizontal Pod Autoscaler de Kubernetes) basado en métricas (CPU >70%, memoria >80%, latencia >umbral) y completarse en <5 minutos.

Cofinanciado por la Unión Europea
Fondos Europeos

Fuente: Pliego HW-L1-3, sección 4.1.3 infraestructura.

ID requisitos relacionados: HW-L1-5, RVE-2a, RVE-2c.

Criterio de aceptación:

- Prueba de carga: iniciar con configuración para carga base (2 réplicas de cada microservicio), generar carga pico con herramienta de testing (JMeter), observar que Kubernetes escala automáticamente réplicas (ej. de 2 a 6 réplicas de API Gateway) en <5 minutos, latencia se mantiene dentro de SLAs.

Prioridad: 5 (requisito explícito pliego).

3.2.4. RNF-INTEROP-04: Tolerancia a desconexión de dispositivos campo

Descripción: Los dispositivos de campo (Lote 2) deben operar autónomamente sin conexión a THOT durante al menos 8 horas (duración típica de intervención), acumulando hasta 500 eventos localmente, y sincronizarlos exitosamente al reconectar sin pérdida de datos ni corrupción.

Fuente: F1.3.1 sección 4.5, pliego INTEROP-7.

ID requisitos relacionados: RF-INTEROP-01.

Criterio de aceptación:

- Prueba de desconexión: dispositivo Lote 2 genera 500 eventos (registro de vestigios, fotografías, eventos CoC) sin conexión, almacena localmente en base de datos SQLite cifrada, reconecta después de 8 horas, sincroniza automáticamente, THOT recibe los 500 eventos, verifica checksums coincidentes, no hay eventos duplicados ni pérdidos.

Prioridad: 5 (requisito explícito pliego).

3.3. Requisitos de seguridad

(Autenticación y autorización entre sistemas (Spring Security / IdP, tokens, certificados); Cifrado de las comunicaciones etc)

Estos requisitos definen controles de seguridad aplicados a integraciones, garantizando confidencialidad, integridad, autenticidad y auditoría.

3.3.1. RS-INTEROP-01: Autenticación mutua (mTLS) en APIs máquina-a-máquina

Descripción: Todas las comunicaciones entre sistemas (THOT ↔ Lote 2, THOT ↔ sistemas TIC internos, THOT ↔ sistemas externos) deben autenticarse mediante **Mutual TLS (mTLS)**: tanto cliente como servidor presentan certificados digitales X.509 emitidos por Autoridad Certificadora (CA) confiable.

Los certificados deben:

- Ser emitidos por CA corporativa de la Policía Nacional o CA del proyecto (a decidir en Fase II).
- Tener vida útil máxima 1 año, con renovación automática antes de expiración.
- Ser revocables (soporte de Certificate Revocation List o OCSP).

Fuente: Pliego 4.3.1.4 seguridad ENS, SEC-L1-3.

ID requisitos relacionados: HW-L1-8.

Criterio de aceptación:

- Configurar API Gateway con política de mTLS obligatorio, intentar petición desde cliente sin certificado válido, verificar rechazo con HTTP 401 Unauthorized.
- Intentar petición con certificado válido, verificar aceptación HTTP 200.
- Revocar certificado de cliente, verificar que peticiones subsiguientes son rechazadas.

Prioridad: 5 (requisito explícito pliego ENS).

3.3.2. RS-INTEROP-02: Cifrado de datos en tránsito

Descripción: Todas las comunicaciones de red entre componentes de THOT y sistemas externos deben cifrarse con **TLS 1.3** (mínimo TLS 1.2 con configuración restringida a cipher suites fuertes: AES-GCM, ChaCha20-Poly1305). Se prohíbe TLS 1.0/1.1 y cipher suites débiles (3DES, RC4, MD5).

Fuente: ENS Alto, pliego HW-L1-8.

ID requisitos relacionados: RS-INTEROP-01.

Criterio de aceptación:

- Analizar tráfico de red con herramienta de inspección (Wireshark), verificar que todos los paquetes entre THOT y Lote 2 están cifrados con TLS 1.3.
- Escanear configuración SSL/TLS de API Gateway con herramienta (testssl.sh), verificar que solo cipher suites fuertes están habilitados.

Prioridad: 5 (requisito explícito ENS Alto).

3.3.3. RS-INTEROP-03: Autorización basada en roles (RBAC) en APIs

Descripción: El acceso a operaciones de API debe estar controlado por **Control de Acceso Basado en Roles (RBAC)**:

- **Rol "Dispositivo Campo"** (Lote 2): puede POST vestigios, GET resultados de análisis propios, subscribirse a alertas propias, no puede acceder a datos de otros casos.
- **Rol "Analista"**: puede GET todos los vestigios de casos asignados, POST resultados de análisis, no puede DELETE vestigios.
- **Rol "Responsable Unidad"**: puede GET todos los vestigios de su unidad territorial, asignar/reasignar análisis, no puede acceder a casos de otras unidades.
- **Rol "Administrador Sistema"**: acceso completo (solo para operaciones técnicas, no acceso rutinario a datos forenses).

Los roles se asignan en el Identity Provider (Keycloak o sistema corporativo) y se validan en cada petición API mediante tokens JWT con claims de roles.

Fuente: ENS Alto, pliego INT-GEN-6.

ID requisitos relacionados: HW-L1-8.

Criterio de aceptación:

- Usuario con rol "Dispositivo Campo" intenta GET vestigio de otro caso, API Gateway rechaza con HTTP 403 Forbidden.
- Usuario con rol "Analista" intenta DELETE vestigio, rechazado con HTTP 403.
- Usuario con rol "Responsable Unidad" intenta GET vestigios de otra unidad, rechazado con HTTP 403.

Prioridad: 5 (requisito explícito ENS Alto).

3.3.4. RS-INTEROP-04: Auditoría completa de transacciones de integración

Descripción: Todas las transacciones de integración (peticiones API, mensajes MQTT, eventos de CoC) deben generar registros de auditoría con información mínima:

- Timestamp (UTC, precisión milisegundos).
- ID de usuario/sistema solicitante (del certificado mTLS o token JWT).
- Operación realizada (ej. POST /api/v1/vestigios, MQTT publish a topic alertas/campo).
- Parámetros relevantes (IDs de recursos afectados, no datos personales completos para cumplir minimización RGPD).
- Resultado (éxito/fallo, código de estado HTTP o MQTT, mensaje de error si aplica).
- IP origen y traza de request-id para correlación distribuida.

Los logs de auditoría deben almacenarse en sistema centralizado inmutable (ej. ElasticSearch con índices write-once), conservarse mínimo 3 años, estar accesibles solo para auditores autorizados y soportar búsquedas eficientes (respuesta <5 segundos para consultas simples).

Fuente: ENS Alto, pliego HW-L1-11.

ID requisitos relacionados: HW-L1-8, INT-GEN-5.

Criterio de aceptación:

- Realizar 100 peticiones API con diferentes usuarios/operaciones, consultar logs de auditoría, verificar que existen 100 registros con información completa (timestamp, usuario, operación, resultado).
- Intentar modificar un log de auditoría directamente en ElasticSearch, verificar que falla (configuración write-once).

Prioridad: 5 (requisito explícito ENS Alto).

3.3.5. RS-INTEROP-05: Protección de datos biométricos en tránsito y reposo

Descripción: Los datos biométricos (huellas dactilares, fotografías faciales, perfiles genéticos) son categorías especiales de datos personales según RGPD. Su transmisión y almacenamiento requiere protección adicional:

- **En tránsito**: cifrado TLS 1.3 (ya cubierto por RS-INTEROP-02) más encriptación adicional a nivel de aplicación para imágenes/archivos grandes (ej. AES-256-GCM antes de envío).
- **En reposo**: almacenamiento en base de datos con cifrado transparente (TDE) o cifrado a nivel de aplicación con claves gestionadas por servicio de gestión de secretos (HashiCorp Vault).
- **Minimización**: solo transmitir/almacenar datos biométricos cuando sea estrictamente necesario; preferir identificadores/hashes cuando sea posible.

Fuente: RGPD artículo 9, pliego PROT-DATOS-1.

ID requisitos relacionados: HW-L1-8, HW-L1-9.

Criterio de aceptación:

- Analizar tráfico de envío de fotografía facial desde Lote 2 a THOT, verificar que imagen está cifrada (no visualizable en packet capture).
- Acceder directamente a base de datos donde se almacenan huellas dactilares, verificar que están cifradas (no legibles en texto plano).

Prioridad: 5 (requisito explícito RGPD).

3.4. Requisitos normativos

(RGPD, ENS, Reglamento PrümII, ISO/IEC 27043, etc.)
Estos requisitos derivan de obligaciones legales y normativas aplicables al ámbito forense y uso de IA.

3.4.1. RN-INTEROP-01: Cumplimiento de ISO 21043 (Cadena de Custodia)

Descripción: El registro de cadena de custodia de evidencias digitales debe cumplir la norma **ISO 21043:2018 "Forensic sciences – Investigation – Principles and requirements"**, específicamente:

- Identificación única e inequívoca de cada evidencia.
- Registro de todas las transferencias de custodia (quién, cuándo, dónde, por qué).
- Registro de todos los accesos a la evidencia (quién accedió, cuándo, qué operación realizó).
- Integridad verificable (checksums, firmas digitales).
- Documentación de procedimientos de almacenamiento, manipulación y transporte.

Fuente: Pliego 4.1.1.2, 4.2.1.1.4, pliego INS-EJE-6.

ID requisitos relacionados: RF-INTEROP-04.

Criterio de aceptación:

- Auditoría de implementación de cadena de custodia por experto externo (ej. auditor ENAC), verificar cumplimiento de requisitos ISO 21043, emitir informe de conformidad.

Prioridad: 5 (requisito explícito pliego).

3.4.2. RN-INTEROP-02: Cumplimiento de EU AI Act (IA de Alto Riesgo)

Descripción: Los componentes de IA de THOT que intervienen en procesos forenses (reconocimiento biométrico, análisis predictivo, recomendación de hipótesis) están clasificados como **sistemas de IA de alto riesgo** según EU AI Act (aplicaciones de identificación biométrica remota, sistemas utilizados en contexto de aplicación de la ley). Esto impone obligaciones:

- **Transparencia**: documentar arquitectura del modelo, datos de entrenamiento, limitaciones conocidas.
- **Explicabilidad**: proporcionar explicaciones comprensibles de decisiones de IA [HW-L1-12].
- **Supervisión humana**: diseñar flujos de trabajo donde decisiones críticas de IA requieren validación humana antes de acción (ej. match biométrico con consecuencias jurídicas requiere revisión por experto).
- **Registro de operaciones**: log de todas las inferencias de IA con inputs, outputs, modelo utilizado [HW-L1-13].
- **Evaluación de riesgos y sesgo**: análisis continuo de equidad del modelo [HW-L1-13].

En el contexto de interoperabilidad, esto aplica especialmente a APIs de IA expuestas a sistemas externos (ej. endpoint de reconocimiento facial consumido por Lote 2).

Cofinanciado por la Unión Europea
Fondos Europeos

Fuente: EU AI Act, pliego HW-L1-12, HW-L1-13, INT-43.

ID requisitos relacionados: RN-INTEROP-03 (Convenio Marco IA).

Criterio de aceptación:

- Documentar modelo de reconocimiento facial utilizado en API, incluyendo: arquitectura del modelo, dataset de entrenamiento (origen, tamaño, diversidad demográfica), métricas de rendimiento desagregadas por etnia/género, limitaciones conocidas.
- Implementar endpoint /api/v1/ia/facial-recognition con parámetro explain=true que retorna explicación XAI (ej. heatmap de regiones faciales relevantes, score de confianza, advertencias).
- Implementar flujo donde resultado de match biométrico con score <umbral configurable requiere revisión manual antes de generar alerta a sistema externo.

Prioridad: 5 (requisito explícito normativa UE aplicable).

3.4.3. RN-INTEROP-03: Cumplimiento de Convenio Marco sobre IA y Derechos Humanos

Descripción: El uso de IA en aplicaciones de seguridad debe respetar derechos fundamentales (dignidad, privacidad, no discriminación) según Convenio Marco del Consejo de Europa sobre IA. En contexto de interoperabilidad, esto implica:

- **No uso de IA para atribución automática de culpabilidad**: los sistemas de IA pueden asistir (ej. sugerir hipótesis, identificar patrones) pero no decidir autónomamente responsabilidad penal.
- **Derecho a explicación**: las personas afectadas por decisiones asistidas por IA tienen derecho a explicación comprensible.
- **Prohibición de perfilado indebido**: no usar IA para crear perfiles discriminatorios basados en características protegidas (etnia, religión, orientación sexual).

En APIs de integración, esto se traduce en:

- No exponer endpoints de IA que permitan perfilado automático sin supervisión.
- Incluir disclaimers en documentación de APIs de IA sobre limitaciones y necesidad de supervisión humana.
- Implementar controles de uso justo (fair use) en APIs de IA para prevenir abuso.

Fuente: Convenio Marco CoE sobre IA, memoria técnica 2.1.8 consideraciones éticas.

ID requisitos relacionados: RN-INTEROP-02, INT-43.

Criterio de aceptación:

- Documentación de API incluye sección "Consideraciones Éticas y Limitaciones" para cada endpoint de IA, explicando: propósito legítimo del modelo, limitaciones conocidas, obligación de supervisión humana, prohibiciones de uso.

- Implementar mecanismo de rate limiting específico para APIs de IA (ej. máximo 100 peticiones/hora por cliente) para prevenir uso masivo no supervisado.

Prioridad: 5 (requisito explícito normativa aplicable).

3.4.4. RN-INTEROP-04: Cumplimiento de RGPD en intercambio de datos personales

Descripción: El intercambio de datos personales (filiación, biométricos, antecedentes) entre THOT y sistemas externos debe cumplir RGPD, especialmente:

- **Licitud del tratamiento**: base jurídica para procesamiento (ej. ejercicio de funciones públicas en aplicación de ley, art. 6.1.e RGPD; tratamiento de categorías especiales para fines de interés público esencial, art. 9.2.g).
- **Minimización**: solo transmitir datos estrictamente necesarios para la finalidad (ej. si sistema externo solo necesita confirmar identidad, enviar solo NIF+hash facial, no fotografía completa).
- **Limitación de finalidad**: datos intercambiados solo pueden usarse para finalidad especificada (investigación criminal), no reutilización para otros fines.
- **Transferencias internacionales**: si se intercambian datos con sistemas fuera UE (ej. INTERPOL en países terceros), aplicar salvaguardas RGPD (decisión de adecuación, cláusulas contractuales tipo).

Fuente: RGPD, pliego PROT-DATOS-1.

ID requisitos relacionados: RS-INTEROP-05.

Criterio de aceptación:

- Documentar base jurídica para cada tipo de intercambio de datos con sistemas externos (ej. tabla: Sistema Destino | Tipo de datos | Base jurídica RGPD | Salvaguardas).
- Implementar política de minimización en adaptadores: ej. adaptador para sistema que solo necesita verificar identidad debe enviar solo NIF+hash, no datos completos de filiación.
- Auditoría RGPD por DPO o auditor externo, verificar conformidad de arquitectura de interoperabilidad, emitir informe.

Prioridad: 5 (requisito explícito RGPD).

3.5. Requisitos de portabilidad

(evitar dependencias propietarias, vendor lock-in)
Estos requisitos garantizan que THOT no está acoplado a infraestructura o proveedores específicos, facilitando evolución futura.

Cofinanciado por la Unión Europea
Fondos Europeos

3.5.1. RP-INTEROP-01: Independencia de infraestructura cloud/on-premise

Descripción: La arquitectura de interoperabilidad de THOT debe ser desplegable tanto en infraestructura on-premise (centros de datos propios de Policía Nacional) como en cloud público (ej. AWS, Azure, GCP) o cloud privado, sin requerir rediseño arquitectónico. Esto se logra mediante:

- Uso de **Kubernetes** como capa de abstracción de infraestructura.
- Almacenamiento basado en **interfaces estándar** (S3 API para object storage, PersistentVolumes de Kubernetes para discos).
- Red basada en **estándares** (CNI de Kubernetes, no plugins propietarios de cloud específico).
- Servicios gestionados reemplazables: ej. RDS (AWS) vs CloudSQL (GCP) vs PostgreSQL autogestionado, todos compatibles con driver PostgreSQL estándar.

Fuente: Pliego HW-L1-4.

ID requisitos relacionados: HW-L1-3.

Criterio de aceptación:

- Desplegar THOT completo en entorno on-premise (Fase II desarrollo), migrar a entorno cloud simulado (ej. Minikube en cloud VM), verificar que funciona sin cambios en código, solo actualización de manifiestos Kubernetes y configuración de infraestructura.

Prioridad: 5 (requisito explícito pliego, necesario para cumplir ENS Alto que puede requerir on-premise).

3.5.2. RP-INTEROP-02: Independencia de proveedores de tecnología (evitar vendor lock-in)

Descripción: Preferir tecnologías open source con comunidades activas y estándares abiertos sobre soluciones propietarias de un único proveedor, para evitar dependencia (vendor lock-in) y asegurar sostenibilidad a largo plazo:

- **API Gateway**: Kong (open source) o Nginx Ingress Controller > soluciones propietarias de cloud específico.
- **Bus de eventos**: Apache Kafka (open source) > servicios propietarios (AWS Kinesis, Azure Event Hubs).
- **Registro inmutable**: ImmuDB (open source) > soluciones propietarias de inmutabilidad/ledger.
- **Identity Provider**: Keycloak (open source) > soluciones propietarias, aunque integrable con IdP corporativo si existe.

Cuando se use tecnología de proveedor específico, asegurar que existe alternativa open source compatible o protocolo estándar (ej. si se usa AWS S3, asegurar compatibilidad con API S3 estándar para poder migrar a MinIO open source).

Fuente: Memoria técnica 2.1.2.8 infraestructura, principio de sostenibilidad.

ID requisitos relacionados: HW-L1-4.

Criterio de aceptación:

- Auditoría de stack tecnológico: verificar que >80% de componentes críticos de interoperabilidad son open source o basados en estándares abiertos.
- Documentar para cada componente propietario (si existe): alternativa open source equivalente, esfuerzo estimado de migración.

Prioridad: 4 (requisito identificado por consorcio, derivado de sostenibilidad).

4. PRINCIPIOS DE DISEÑO

Esta sección establece los principios arquitectónicos fundamentales que guían todas las decisiones de diseño e implementación de la arquitectura de interoperabilidad de THOT. Estos principios son vinculantes: cualquier decisión técnica que los contradiga debe ser justificada explícitamente y documentada como excepción.

4.1. Principios de arquitectura

(SOA, microservicios, event-driven, API-first, etc.)

4.1.1. P1: Modularidad y desacoplamiento

Enunciado: La arquitectura de interoperabilidad se organiza en componentes modulares independientes que interactúan mediante interfaces bien definidas, minimizando dependencias directas entre módulos.

Justificación: Permite evolucionar, sustituir o escalar componentes individuales sin afectar al sistema completo. Facilita pruebas unitarias, despliegue independiente y distribución del trabajo entre equipos.

Aplicación concreta:

- Cada tipo de integración (Lote 2, sistemas TIC internos, sistemas externos) se implementa como **microservicio independiente** con su propia base de datos (patrón Database per Service).
- Los microservicios se comunican mediante **APIs REST síncronas** (para operaciones request-response) o **mensajería asíncrona** vía Apache Kafka (para eventos y notificaciones), nunca mediante llamadas directas a bases de datos de otros servicios.
- El **API Gateway** actúa como fachada única, desacoplando clientes externos de la implementación interna de microservicios (permite cambiar enrutamiento, añadir microservicios, sin afectar clientes).

Verificación: En revisiones de diseño, aplicar métrica de acoplamiento: ningún microservicio debe tener dependencias directas (imports de código, acceso a BBDD) de más de 2 otros microservicios. Dependencias adicionales deben mediarse por bus de eventos.

Requisitos relacionados: HW-L1-4, INTEROP-3.

4.1.2. P2: Orientación a servicios (SOA)

Enunciado: Las capacidades de interoperabilidad se exponen como servicios reutilizables, cohesivos (una responsabilidad bien definida) y descubribles (documentación accesible, contratos explícitos).

Justificación: Maximiza reutilización, facilita composición de funcionalidades complejas a partir de servicios básicos, simplifica gobierno de APIs.

Aplicación concreta:

- Cada servicio de interoperabilidad (ej. GestionVestigiosService, CadenaCustodiaService, BiometricMatchingService) se expone mediante **API REST** con especificación OpenAPI 3.0 publicada en portal de desarrolladores.
- Los servicios son **stateless**: no mantienen estado de sesión entre peticiones, toda la información necesaria viaja en el request (más tokens JWT para autenticación). Esto permite balanceo de carga transparente y escalado horizontal.
- Cada servicio implementa **health check endpoint** (GET /health) que retorna estado del servicio (healthy/unhealthy) + dependencias (BBDD, sistemas externos), permitiendo orquestación automática (Kubernetes liveness/readiness probes).

Verificación: Auditoría de servicios: cada servicio debe tener especificación OpenAPI completa, health check implementado, métricas expuestas (formato Prometheus), logs estructurados (JSON).

Requisitos relacionados: INTEROP-3, HW-L1-4.

4.1.3. P3: Interoperabilidad basada en estándares

Enunciado: Preferir estándares abiertos e internacionalmente reconocidos sobre protocolos propietarios, para maximizar interoperabilidad presente y futura.

Justificación: Reduce riesgo de obsolescencia tecnológica, facilita integración con sistemas externos que también siguen estándares, amplía ecosistema de herramientas/librerías disponibles.

Aplicación concreta:

- **Protocolos de comunicación**: HTTP/REST (RFC 7231), MQTT v5.0 (OASIS standard), gRPC (protocolo abierto Google), WebRTC (W3C standard).
- **Formatos de datos**: JSON (RFC 8259), JSON Schema (draft-07), JSON-LD para semántica (W3C standard), XML cuando sea necesario (ej. integración con sistemas legacy SOAP).
- **Autenticación/autorización**: OAuth 2.0 (RFC 6749), OpenID Connect, JWT (RFC 7519), mTLS (RFC 8446).
- **Semántica**: Ontologías basadas en RDF/OWL (W3C), vocabularios controlados (ej. Dublin Core para metadatos).
- **Cadena de custodia**: ISO 21043:2018 como referencia normativa.

Verificación: En cada decisión de protocolo/formato, documentar: ¿qué estándar se aplica?, ¿qué versión?, ¿hay alternativas propietarias descartadas y por qué?

Requisitos relacionados: INTEROP-1, INTEROP-3, RN-INTEROP-01.

4.1.4. P4: Seguridad por diseño (Security by Design)

Enunciado: Los controles de seguridad se integran desde la concepción del diseño, no como capa añadida posteriormente. Todo flujo de datos asume entorno hostil (principio de "zero trust").

Justificación: Reduce vulnerabilidades, facilita cumplimiento normativo (ENS Alto), aumenta confianza de usuarios y auditores.

Cofinanciado por la Unión Europea
Fondos Europeos

Aplicación concreta:

- **Autenticación obligatoria**: ninguna API se expone sin autenticación (mTLS para máquina-a-máquina, OAuth 2.0 + OpenID Connect para humanos).
- **Autorización granular**: control de acceso basado en roles (RBAC) aplicado en API Gateway + microservicios (defensa en profundidad), principio de mínimo privilegio.
- **Cifrado ubicuo**: TLS 1.3 para datos en tránsito, AES-256 para datos en reposo (bases de datos, object storage), claves gestionadas por servicio dedicado (HashiCorp Vault o equivalente).
- **Auditoría completa**: todas las operaciones de interoperabilidad generan log auditable con información de identidad, operación, timestamp, resultado, almacenado en sistema inmutable.
- **Validación de entrada**: todo dato recibido desde exterior se valida contra esquema JSON Schema antes de procesamiento (prevención de inyección de código, datos malformados).

Verificación: Realizar threat modeling (ej. STRIDE) de cada flujo de integración, identificar amenazas, documentar mitigaciones implementadas. Pruebas de penetración en Fase II.

Requisitos relacionados: SEC-L1-3, HW-L1-8, RS-INTEROP-01 a RS-INTEROP-05.

4.1.5. P5: Escalabilidad horizontal

Enunciado: La capacidad de procesamiento de integraciones debe poder incrementarse añadiendo más instancias de componentes (escalar "hacia fuera"), no solo aumentando recursos de instancias existentes (escalar "hacia arriba").

Justificación: Escalar horizontalmente es más eficiente en costo (usar múltiples máquinas pequeñas en lugar de una máquina enorme), más resiliente (fallo de una instancia no tumba el servicio) y se alinea con arquitectura cloud-native.

Aplicación concreta:

- Todos los microservicios de interoperabilidad son **stateless** (no mantienen estado en memoria entre peticiones), permitiendo balanceo de carga round-robin entre múltiples réplicas.
- Bases de datos diseñadas para escalabilidad: PostgreSQL con particionamiento horizontal (sharding) cuando sea necesario, MongoDB con replica sets, Redis Cluster para caché distribuida.
- Apache Kafka escala horizontalmente añadiendo brokers y particionando topics según carga.
- Kubernetes **Horizontal Pod Autoscaler (HPA)** configurado para escalar automáticamente réplicas de microservicios cuando CPU >70% o latencia >umbral.

Verificación: Pruebas de carga (sección 3.2 RNF-INTEROP-03): incrementar carga desde baseline a 5x, observar escalado automático, verificar que latencia se mantiene dentro de SLAs.

Requisitos relacionados: HW-L1-3, HW-L1-5, RVE-2a, RVE-2c.

4.1.6. P6: Resiliencia y tolerancia a fallos

Enunciado: El sistema debe continuar operando (posiblemente con funcionalidad degradada) ante fallos parciales de componentes, redes o sistemas externos.

Justificación: Incrementa disponibilidad percibida por usuarios (crítico en operaciones policiales 24/7), reduce impacto de fallos en cascada, facilita mantenimiento sin ventanas de downtime.

Aplicación concreta:

- **Redundancia**: múltiples réplicas de cada microservicio crítico (mínimo 3), desplegadas en diferentes nodos físicos/zonas de disponibilidad si la infraestructura lo soporta.
- **Circuit breakers**: si un sistema externo (ej. ABIS) falla repetidamente, el adaptador abre "circuito" (deja de intentar llamadas) temporalmente, retorna error inmediato a clientes, reintenta periódicamente hasta que sistema externo se recupere. Implementado con librerías (ej. Resilience4j en Java/Spring).
- **Timeouts y retries**: todas las llamadas a sistemas externos tienen timeout configurado (ej. 10 segundos), si falla se reintenta con backoff exponencial (1s, 2s, 4s...), máximo 3 intentos, luego se reporta error.
- **Degradación elegante**: si sistema externo no está disponible, THOT puede operar con funcionalidad reducida (ej. si ABIS no responde, permitir registro de vestigio pero marcar cotejo biométrico como "pendiente", procesar cuando ABIS se recupere).
- **Colas persistentes**: eventos críticos (ej. registro de cadena de custodia) se encolan en Kafka (persistente, replicado) antes de procesamiento, garantizando que no se pierden ante fallo de procesador.

Verificación: Chaos engineering en Fase II: inyectar fallos controlados (matar pods aleatorios, simular latencia de red, desconectar sistema externo mock), observar que THOT continúa operando, medir impacto en disponibilidad/latencia.

Requisitos relacionados: HW-L1-7, RNF-INTEROP-02, RNF-INTEROP-04.

4.1.7. P7: Observabilidad y trazabilidad

Enunciado: El comportamiento del sistema de interoperabilidad debe ser observable en tiempo real mediante métricas, logs y trazas distribuidas, facilitando diagnóstico de problemas y optimización de rendimiento.

Justificación: En sistemas distribuidos complejos, es imposible predecir todos los modos de fallo. La observabilidad permite detectar, diagnosticar y resolver incidentes rápidamente.

Aplicación concreta:

- **Métricas**: cada microservicio expone métricas en formato Prometheus (endpoint /metrics): tasa de peticiones, latencia (percentiles 50/95/99), tasa de error, recursos (CPU/memoria), métricas de negocio (ej. número de vestigios registrados/hora). Métricas se centralizan en Prometheus, se visualizan en Grafana con dashboards predefinidos.
- **Logs estructurados**: todos los logs se emiten en formato JSON con campos estándar (timestamp UTC, nivel, servicio, traceId, userId, mensaje, contexto), se centralizan en ElasticSearch (vía Fluentd/Logstash), se consultan en Kibana.

- **Trazas distribuidas**: cada petición de cliente se asigna un traceId único (propagado en header HTTP X-Trace-Id), se registra en cada microservicio por el que pasa, se visualiza en Jaeger permitiendo ver flujo completo end-to-end y detectar cuellos de botella.
- **Alertas proactivas**: configurar alertas en Prometheus (ej. "latencia p95 > 1s durante 5 minutos") que notifican a equipo de operaciones vía sistema de alertas (INT-17).

Verificación: En pruebas de integración, generar petición compleja (ej. registro de vestigio que dispara cotejo biométrico), consultar Jaeger, verificar traza completa visible con tiempos de cada microservicio. Simular fallo, verificar que alertas se disparan.

Requisitos relacionados: HW-L1-7, HW-L1-11, INT-17.

4.2. Estándares y normas de referencia

(RESTful, JSON, XML, OAuth2/OIDC, HL7, NIIF, EUREKA, frameworks de seguridad, etc)
Esta sección enumera los estándares técnicos y normas de calidad/seguridad que se aplican a la arquitectura de interoperabilidad de THOT, indicando su ámbito de aplicación y nivel de cumplimiento (obligatorio/recomendado).

4.2.1. Estándares de protocolos y formatos

Estándar	Versión	Ámbito de aplicación	Nivel	Fuente
HTTP/1.1, HTTP/2, HTTP/3	RFC 7540, RFC 9114	APIs REST, comunicaciones sincrónicas	Obligatorio	INTEROP-3
REST (Representational State Transfer)	Estilo arquitectónico	Diseño de APIs	Obligatorio	INTEROP-3
OpenAPI Specification	3.0	Documentación de APIs REST	Obligatorio	INTEROP-3
JSON (JavaScript Object Notation)	RFC 8259	Formato de intercambio de datos	Obligatorio	INTEROP-1

Estándar	Versión	Ámbito de aplicación	Nivel	Fuente
JSON Schema	Draft-07	Validación de esquemas de datos	Obligatorio	RCD-INTEROP-01
JSON-LD (JSON for Linking Data)	W3C Recommendation 1.1	Enriquecimiento semántico de datos	Recomendado	INTEROP-4
MQTT (Message Queuing Telemetry Transport)	v5.0 OASIS	Mensajería asíncrona desde campo	Obligatorio (Lote 2)	F1.3.1
gRPC	Protocolo abierto	Comunicaciones RPC eficientes	Recomendado	Memoria técnica
WebRTC	W3C Standard	Streaming de vídeo/audio tiempo real	Obligatorio (asistencia remota)	F1.3.1 sección 8
XML	W3C Recommendation	Integración con sistemas legacy SOAP	Condicional (si necesario)	INT-GEN-8

Tabla 3. Estándares de protocolos y formatos de interoperabilidad adoptados por THOT. *Enumera los estándares aplicables, indicando su versión, el ámbito de aplicación y el nivel*

4.2.2. Estándares de seguridad y autenticación

Estándar	Versión	Ámbito de aplicación	Nivel	Fuente
TLS (Transport Layer Security)	1.3 (mínimo 1.2)	Cifrado de comunicaciones	Obligatorio	RS-INTEROP-02, ENS Alto
mTLS (Mutual TLS)	RFC 8446	Autenticación máquina-a-máquina	Obligatorio	RS-INTEROP-01, ENS Alto

Cofinanciado por la Unión Europea · Fondos Europeos

Estándar	Versión	Ámbito de aplicación	Nivel	Fuente
OAuth 2.0	RFC 6749	Autorización delegada	Obligatorio (humanos)	SEC-L1-3
OpenID Connect	1.0	Autenticación de usuarios	Obligatorio (humanos)	SEC-L1-3
JWT (JSON Web Tokens)	RFC 7519	Tokens de autenticación/autorización	Obligatorio	SEC-L1-3
X.509	ITU-T Recommendation	Certificados digitales	Obligatorio (mTLS)	RS-INTEROP-01
SAML 2.0	OASIS	Federación de identidades	Condicional (si IdP corporativo usa)	-

Tabla 4. Estándares de seguridad y autenticación aplicables a las integraciones de THOT. *Enumera los estándares aplicables, indicando su versión, el ámbito de aplicación y el nivel.*

4.2.3. Normas de calidad y gestión forense

Norma	Versión	Ámbito de aplicación	Nivel	Fuente
ISO 21043	2018	Principios de investigación forense y cadena de custodia	Obligatorio	RN-INTEROP-01, pliego 4.1.1.2
ISO/IEC 17025	2017	Requisitos generales para competencia de laboratorios de ensayo	Obligatorio (THOT como soporte a labs)	Pliego 2.1
ISO 17020	2012	Requisitos para organismos de inspección	Obligatorio (ITP)	Pliego 2.1
ISO/IEC 27001	2013	Sistemas de gestión de seguridad de la información	Obligatorio (ENS Alto)	SEC-L1-3
ISO/IEC 27002	2022	Controles de seguridad de la información	Obligatorio (ENS Alto)	SEC-L1-3

Norma	Versión	Ámbito de aplicación	Nivel	Fuente
ISO/IEC 27701	2019	Gestión de privacidad de la información (GDPR)	Obligatorio	PROT-DATOS-1

Tabla 5. Normas de calidad y gestión forense. *Enumera las normas aplicables, indicando su versión, el ámbito de aplicación y el nivel.*

4.2.4. Regulación y normativa aplicable

Regulación	Ámbito de aplicación	Nivel	Fuente
RGPD (Reglamento General de Protección de Datos)	Tratamiento de datos personales y biométricos	Obligatorio	RN-INTEROP-04, pliego 4.3.1.2
ENS (Esquema Nacional de Seguridad)	Nivel Alto	Seguridad de sistemas de información del sector público	Obligatorio
EU AI Act	Sistemas de IA de alto riesgo (biometría, seguridad)	Obligatorio	RN-INTEROP-02, pliego 2.1.2
Convenio Marco CoE sobre IA	Uso responsable de IA en seguridad pública	Obligatorio	RN-INTEROP-03
Directiva NIS2	Ciberseguridad de entidades críticas	Aplicable (Policía Nacional como entidad crítica)	Pliego
Reglamento Prüm II	Intercambio automatizado de datos biométricos UE	Aplicable (integración futura)	Pliego 4.1.2, INTEROP-2

Tabla 6. Regulación y normativa aplicable al intercambio de datos y el uso de IA*. Resume el marco legal exigible o aplicable y su impacto sobre diseño y operación.*

4.2.5. Estándares de desarrollo y operación

Estándar/Práctica	Ámbito de aplicación	Nivel	Fuente
OpenAPI Specification 3.0	Documentación de APIs REST	Obligatorio	INTEROP-3

Estándar/Práctica	Ámbito de aplicación	Nivel	Fuente
AsyncAPI 2.x	Documentación de APIs asíncronas (MQTT, Kafka)	Recomendado	-
Semantic Versioning (SemVer)	Versionado de APIs y componentes	Obligatorio	Sección 6.3
RFC 7807 (Problem Details for HTTP APIs)	Formato de errores en APIs REST	Recomendado	F1.3.1 sección 5
Prometheus exposition format	Formato de métricas	Obligatorio	Principio P7
OpenTelemetry	Instrumentación de observabilidad	Recomendado	Principio P7
BPMN 2.0	Modelado de procesos de negocio	Obligatorio (flujos de trabajo)	Memoria técnica 2.1.2.4

Tabla 7. Estándares de desarrollo y operación. *Enumera los estándares y prácticas aplicables, indicando el ámbito de aplicación y el nivel (obligatorio/recomendado).*

4.2.6. Aplicación de estándares en fases del proyecto

- **Fase I (Diseño)**: Selección de estándares aplicables, documentación de especificaciones, diseño de interfaces conformes.
- **Fase II (Desarrollo)**: Implementación conforme a estándares, uso de librerías certificadas/validadas, pruebas de conformidad.
- **Fase III (Validación)**: Auditoría de cumplimiento por entidad externa (ej. auditor ENS, certificador ISO), corrección de desviaciones, obtención de certificaciones (si aplica).

Gestión de conflictos entre estándares: En caso de que dos estándares aplicables sean contradictorios (ej. recomendación de un estándar técnico vs obligación normativa), prevalece la **obligación normativa** (ENS, RGPD, EU AI Act).

4.3. Criterios de extensibilidad y escalabilidad

Esta sección define criterios específicos que garantizan que la arquitectura de interoperabilidad puede evolucionar y crecer sin requerir rediseño fundamental.

4.3.1. Criterio E1: Adición de nuevos sistemas externos sin cambio de núcleo

Definición: Debe ser posible integrar un nuevo sistema externo (ej. nueva base de datos policial, nuevo sistema forense internacional) añadiendo un **microservicio adaptador** nuevo, sin modificar componentes centrales de THOT (API Gateway, bus de eventos, servicios de negocio).

Mecanismo de implementación:

- Definir **interface** **abstracta** ExternalSystemAdapter con métodos: connect(), authenticate(), query(), submit(), disconnect().
- Cada nuevo sistema externo implementa esta interfaz como microservicio independiente.
- El microservicio se registra en el **Service Registry** (ej. Consul, Eureka, o service discovery de Kubernetes), indicando qué tipo de sistema maneja (ej. "biometric-database", "forensic-lab").
- Los servicios de negocio que necesitan consultar sistemas externos usan el **patrón Service Locator**: consultan el Service Registry para obtener instancias de adaptadores del tipo necesario, invocan mediante interfaz abstracta.

Criterio de aceptación: En Fase II, añadir adaptador mock para un sistema externo no implementado inicialmente (ej. sistema INTERPOL simulado) en <1 semana, sin modificar líneas de código de servicios existentes, solo añadiendo nuevo microservicio + configuración en Service Registry.

Requisitos relacionados: HW-L1-4, RF-INTEROP-03.

4.3.2. Criterio E2: Versionado de APIs compatible hacia atrás

Definición: Las actualizaciones de APIs REST expuestas por THOT deben permitir que clientes existentes (dispositivos Lote 2, aplicaciones de terceros) continúen funcionando sin modificación durante un **periodo de transición mínimo de 6 meses**.

Mecanismo de implementación:

- Versionado de APIs mediante **URL**: /api/v1/vestigios, /api/v2/vestigios. Ambas versiones coexisten.
- **Cambios compatibles** (no requieren nueva versión): añadir campos opcionales, añadir nuevos endpoints, añadir valores a enums.
- **Cambios incompatibles** (requieren nueva versión): eliminar/renombrar campos, cambiar tipos de datos, cambiar semántica de operaciones.
- Cuando se introduce v2, v1 se marca como **deprecated** (se documenta en OpenAPI, se retorna header HTTP Warning: 299 - "API v1 deprecated, migrate to v2 by YYYY-MM-DD"), pero continúa funcionando 6 meses. Tras 6 meses, v1 se elimina (retorna HTTP 410 Gone).

Criterio de aceptación: Documentar en especificación OpenAPI política de versionado, implementar pruebas automatizadas que verifiquen que v1 y v2 coexisten y retornan respuestas válidas (aunque diferentes) para misma petición.

Requisitos relacionados: Sección 6.3, HW-L1-4.

4.3.3. Criterio E3: Escalabilidad de almacenamiento

Definición: Las bases de datos utilizadas para almacenar datos de interoperabilidad (logs, cadena de custodia, metadatos de vestigios) deben poder crecer desde dataset inicial (estimado: 10 GB, 100K registros) hasta dataset futuro (estimado: 10 TB, 100M registros) sin degradación significativa de rendimiento (<20% incremento en latencia de consultas).

Mecanismo de implementación:

- **Particionamiento (sharding)**: bases de datos relacionales (PostgreSQL) se particionan horizontalmente por clave (ej. asuntoId o timestamp), cada partición en servidor separado.
- **Índices optimizados**: crear índices en campos de consulta frecuente (ej. índice B-tree en vestigioId, índice GIN en campos full-text, índice BRIN en timestamps para range queries).
- **Archivado de datos históricos**: datos de casos cerrados >5 años se migran a almacenamiento "frío" (ej. S3 Glacier), accesibles pero con mayor latencia.
- **Caché distribuida**: consultas frecuentes se cachean en Redis Cluster (ttl configurable), reduciendo carga en base de datos principal.

Criterio de aceptación: Pruebas de carga en Fase II con dataset sintético de 1M registros, medir latencia de consultas típicas (percentil 95), extrapolar a 100M registros usando modelo de escalabilidad, verificar que latencia proyectada <3 segundos.

Requisitos relacionados: HW-L1-3, INTEROP-4.

4.3.4. Criterio E4: Escalabilidad de procesamiento de eventos

Definición: El bus de eventos (Apache Kafka) debe soportar incremento desde carga base (500 eventos/segundo) hasta carga pico (5000 eventos/segundo) sin pérdida de eventos ni incremento de latencia >50%.

Mecanismo de implementación:

- **Particionamiento de topics**: cada topic de Kafka (ej. vestigios.creados, analisis.completados) se particiona en N particiones (inicialmente N=10), permitiendo procesamiento paralelo por N consumidores.
- **Replicación**: cada partición se replica en mínimo 3 brokers de Kafka, garantizando durabilidad.
- **Escalado horizontal de brokers**: añadir más brokers de Kafka incrementa capacidad de procesamiento, las particiones se redistribuyen automáticamente (Kafka rebalancing).
- **Consumer groups**: múltiples instancias de cada microservicio consumidor se organizan en consumer group, Kafka distribuye particiones entre instancias, escalando procesamiento.

Criterio de aceptación: Prueba de carga en Fase II: generar 5000 eventos/segundo en Kafka durante 10 minutos, medir: (1) 0% pérdida de eventos (verificar contador de eventos producidos vs consumidos), (2) latencia end-to-end (tiempo desde producción de evento hasta procesamiento por consumidor) percentil 95 <2 segundos.

Requisitos relacionados: HW-L1-5, RNF-INTEROP-03.

4.4. Gobernanza interlotes

Esta sección define el modelo de gobernanza técnica entre THOT (Lote 1) y las soluciones del Lote 2, asegurando coordinación en diseño, desarrollo y evolución de las interfaces compartidas.

4.4.1. Estructura de gobernanza

Comité Técnico de Interoperabilidad (CTI): Órgano de coordinación técnica entre ambos lotes, compuesto por:

- Arquitecto técnico principal de THOT (UTE ForensIA THOT).
- Arquitecto técnico principal de Lote 2 (adjudicatario Lote 2).
- Responsable técnico de la Policía Nacional (usuario final).
- Representante del CDTI (supervisión contractual).

Responsabilidades del CTI:

- Aprobar cambios en interfaces compartidas (APIs, esquemas de datos, protocolos) definidas en F1.3.1.
- Resolver conflictos técnicos entre lotes (ej. ambigüedades en especificación, limitaciones técnicas descubiertas en implementación).
- Coordinar calendario de despliegue de cambios (asegurar sincronización de versiones entre lotes).
- Aprobar excepciones a estándares cuando exista justificación técnica sólida.

Periodicidad: Reuniones quincenales durante Fase II (desarrollo), mensuales durante Fase III (validación).

4.4.2. Proceso de gestión de cambios en interfaces

Cualquier cambio en las interfaces compartidas entre Lote 1 y Lote 2 (definidas en F1.3.1) sigue el siguiente proceso:

1. **Propuesta de cambio**: Cualquier lote puede proponer cambio documentando: (1) qué se quiere cambiar y por qué, (2) impacto en el otro lote, (3) alternativas consideradas, (4) timeline propuesto. Se envía a CTI.
2. **Revisión técnica**: CTI evalúa propuesta en reunión, puede solicitar aclaraciones o análisis adicional.
3. **Decisión**: CTI aprueba (por consenso preferentemente, por mayoría si es necesario) o rechaza cambio. Si se aprueba, se asigna versión (ej. protocolo v1.1).
4. **Implementación coordinada**: Ambos lotes implementan cambio en sus entornos de desarrollo, prueban integración en entorno conjunto.
5. **Despliegue sincronizado**: Se acuerda fecha de despliegue en preproducción y luego producción, ambos lotes despliegan simultáneamente para evitar incompatibilidades.

Cambios de emergencia: Si se descubre un defecto crítico de seguridad o funcional que requiere cambio inmediato, se puede activar proceso acelerado (decisión en 48 horas) con reunión extraordinaria del CTI.

Registro de cambios: Todos los cambios aprobados se registran en **Change Log** mantenido en repositorio compartido (ej. Git), versionado junto con especificación F1.3.1 actualizada.

4.4.3. Entornos de integración compartidos

Para facilitar desarrollo y pruebas de integración entre lotes, se establecen entornos compartidos:

Entorno de desarrollo conjunto (DEV): Instancias de THOT y Lote 2 desplegadas en infraestructura de desarrollo, accesibles mutuamente, con datos sintéticos. Ambos lotes pueden desplegar cambios libremente (sin aprobación previa), pero se comunican proactivamente vía canal técnico (ej. Slack compartido).

Entorno de preproducción (PRE): Instancias estables de ambos lotes, desplegadas con configuración similar a producción, con datos de prueba realistas (anonimizados). Cambios requieren coordinación (no despliegue sorpresa que rompa integración).

Entorno de producción (PROD): Instancias finales, acceso controlado, despliegues solo tras validación en PRE y aprobación de CTI + Policía Nacional.

4.4.4. Soporte técnico mutuo

Cada lote designa **punto de contacto técnico** (technical liaison) disponible para consultas del otro lote:

- Horario de disponibilidad: L-V 9:00-18:00 (al menos, soporte extendido en fases críticas).
- Canal de comunicación: correo electrónico + Slack/Teams compartido para consultas urgentes.
- SLA de respuesta: consultas no urgentes <24 horas, urgentes <4 horas, críticas (bloqueante para integración) <1 hora.

4.4.5. Resolución de conflictos

Si CTI no logra consenso en una decisión técnica crítica:

- **Escalamiento nivel 1**: Directores de proyecto de ambos lotes intentan resolver bilateralmente.
- **Escalamiento nivel 2**: Policía Nacional (como cliente) toma decisión final basándose en criterio de máximo beneficio operativo.
- **Escalamiento nivel 3** (solo si impacto contractual): CDTI arbitra según términos del pliego.

Toda resolución de conflictos se documenta en acta formal y se añade al registro de decisiones arquitectónicas (ADR).

5. ARQUITECTURA OBJETIVO DE INTEROPERABILIDAD

Esta sección presenta la arquitectura de interoperabilidad de THOT desde cinco vistas complementarias, siguiendo el modelo de vistas arquitectónicas 4+1. Cada vista se enfoca en un aspecto específico, facilitando comprensión por diferentes stakeholders.

5.1. Vista lógica de la arquitectura

(componentes y relaciones entre microservicios, enrutamiento)
La vista lógica describe los componentes funcionales del sistema de interoperabilidad, sus responsabilidades y las relaciones entre ellos, abstrayendo detalles de implementación y despliegue físico.

5.1.1. Componentes principales de interoperabilidad

5.1.1.1. API Gateway (Kong / Nginx Ingress)

Responsabilidad: Punto de entrada único para todas las peticiones externas a THOT (desde Lote 2, sistemas TIC internos, aplicaciones web). Actúa como proxy reverso, aplicando políticas transversales.

Funcionalidades:

- **Enrutamiento**: dirige peticiones a microservicio backend apropiado según URL/método HTTP.
- **Autenticación**: valida tokens JWT (para usuarios humanos) o certificados mTLS (para sistemas máquina-a-máquina), rechaza peticiones no autenticadas.
- **Autorización básica**: aplica políticas de rate limiting (límite de peticiones por cliente), geo-blocking si es necesario (restringir acceso desde IPs no autorizadas).
- **Logging y auditoría**: registra toda petición (timestamp, IP origen, usuario/sistema, endpoint, código de respuesta) en log centralizado.
- **Versionado de APIs**: gestiona coexistencia de múltiples versiones de API (ej. /api/v1/ y /api/v2/), enruta a implementaciones correspondientes.
- **Transformación de protocolos** (opcional): puede transformar peticiones (ej. REST a SOAP para sistemas legacy).

Tecnología: Kong (open source, extensible con plugins Lua) o Nginx Ingress Controller (más ligero, integrado nativamente con Kubernetes). Decisión final en Fase II según requisitos de rendimiento y complejidad de políticas.

Interfaz de entrada: HTTP/HTTPS (puerto 443), mTLS obligatorio para clientes externos.

Interfaz de salida: HTTP hacia microservicios backend (comunicación interna, puede ser HTTP plano dentro de cluster Kubernetes si se confía en aislamiento de red, o mTLS si paranoia de seguridad).

Escalabilidad: múltiples instancias del API Gateway (mínimo 3) detrás de balanceador de carga (ej. HAProxy, AWS NLB), Kubernetes HPA escala automáticamente según CPU.

5.1.1.2. Servicio de Espacio de Datos (Data Space Service)

Responsabilidad: Centralizar y normalizar acceso a datos forenses almacenados en THOT y en sistemas externos, proporcionando capa de abstracción que oculta heterogeneidad de fuentes.

Subfuncionalidades:

- **Servicio de Consulta Unificada**: recibe consultas en formato canónico (ej. "buscar persona por NIF"), traduce a formato específico de sistema fuente (ej. query SQL a BBDD PERSONAS, API REST a sistema ABIS), ejecuta, normaliza resultados a formato canónico THOT.
- **Caché distribuida** (Redis Cluster): cachea resultados de consultas frecuentes a sistemas externos (ej. datos de filiación de personas), reduce latencia y carga en sistemas origen. TTL configurable (ej. 1 hora para datos estables, 5 minutos para datos volátiles).
- **Registro de consultas**: audita todas las consultas a sistemas externos (quién, qué, cuándo) para cumplimiento RGPD y trazabilidad forense.

Adaptadores por sistema externo: Cada sistema externo integrado tiene microservicio adaptador específico (componentes separados lógicamente, pueden desplegarse juntos o separados según carga):

- **Adaptador PERSONAS**: traduce consultas de THOT (formato JSON canónico) a formato específico de sistema PERSONAS (puede ser SQL, SOAP, REST según implementación de PERSONAS), traduce respuestas a JSON canónico.
- **Adaptador ABIS**: similar, específico para sistema biométrico.
- **Adaptador EURODAC**, **Adaptador INTERPOL**, etc. (a implementar progresivamente).

Tecnología: Microservicios en Spring Boot (Java) o FastAPI (Python), comunicación con API Gateway vía REST, comunicación con sistemas externos según protocolo de cada sistema, uso de librerías de resiliencia (Resilience4j) para circuit breakers y retries.

Interfaz de entrada: APIs REST internas (no expuestas directamente a exterior, solo accesibles por otros microservicios de THOT).

Interfaz de salida: Diversa según sistema externo (REST, SOAP, SQL, gRPC...).

5.1.1.3. Bus de Eventos (Apache Kafka)

Responsabilidad: Gestionar comunicación asíncrona basada en eventos entre microservicios de THOT, desacoplando productores de consumidores y permitiendo procesamiento asíncrono resiliente.

Casos de uso:

- **Notificación de eventos de negocio**: cuando microservicio A realiza operación relevante (ej. vestigio creado, análisis completado, alerta generada), publica evento en topic de Kafka. Microservicios interesados (B, C, D) se suscriben a ese topic y procesan el evento de forma independiente.
- **Integración con Lote 2**: eventos generados en campo (captura de vestigio, actualización CoC) se publican en topics de Kafka, microservicios de THOT los consumen para procesamiento.
- **Procesamiento de flujos de datos** (stream processing): análisis en tiempo real de flujos de eventos para detectar patrones (ej. Kafka Streams o Apache Flink consumiendo de Kafka).

Topics principales (lista no exhaustiva):

- vestigios.creados: evento publicado cada vez que se registra nuevo vestigio.
- vestigios.actualizados: cambios en vestigios existentes.
- analisis.completados: resultados de análisis disponibles.
- alertas.generadas: nuevas alertas para notificar.
- coc.eventos: eventos de cadena de custodia (transferencia, acceso).

Configuración:

- Cada topic con mínimo 10 particiones (permite paralelismo de procesamiento).
- Replicación factor 3 (cada partición replicada en 3 brokers).
- Retención de mensajes: 7 días para topics transaccionales, 30 días para topics de auditoría.

Tecnología: Apache Kafka (cluster de mínimo 3 brokers para alta disponibilidad), Zookeeper (coordinación de cluster, puede migrar a KRaft mode sin Zookeeper en futuras versiones).

Interfaz: Protocolo nativo de Kafka, clientes en múltiples lenguajes (Java, Python, Go).

5.1.1.4. Servicio de Cadena de Custodia (Registro Inmutable)

Responsabilidad: Registrar de forma inmutable todos los eventos de cadena de custodia de evidencias digitales, garantizando trazabilidad forense completa y auditoría verificable.

Arquitectura interna:

- **ImmuDB**: servicio de registro inmutable desplegado en infraestructura de THOT. Despliegues participantes: (1) THOT central, (2) Unidades territoriales (si despliegan instancias propias, a decidir en Fase II), (3) Potencialmente Lote 2 (si dispositivos de campo tienen capacidad de interactuar directamente con el servicio, o delegan en THOT central).
- **Validaciones de eventos:** lógica de validación de eventos de custodia implementada en el microservicio (Go o Node.js). Ejemplo de reglas: (1) solo usuario con rol "Técnico Policía Científica" puede registrar transferencia de vestigio, (2) timestamp de evento debe ser posterior a timestamp de evento previo, (3) checksum de archivo asociado a vestigio debe coincidir con valor registrado en evento de creación.
- **Nodos (instancias):** servidores que mantienen copia del registro. Mínimo 3 instancias para tolerancia a fallos.
- **Gestión de identidades:** gestiona identidades digitales de participantes (certificados X.509), integrado con sistema de identidad corporativo de Policía Nacional si existe.

Microservicio Ledger CoC: Capa de abstracción REST sobre ImmuDB, facilita interacción con el registro inmutable sin que clientes necesiten conocer detalles internos.

Funcionalidades del microservicio:

- **POST /api/v1/coc/eventos**: registrar nuevo evento de custodia (ej. transferencia de vestigio). Valida datos, registra el evento en el registro inmutable, retorna ID de transacción.

- **GET /api/v1/coc/vestigio/{id}/historial**: consultar historial completo de eventos de custodia de un vestigio. Consulta el registro inmutable, retorna lista de eventos con timestamps, usuarios, operaciones.
- **GET /api/v1/coc/evento/{txId}**: consultar detalles de un evento específico por ID de transacción.

Sincronización offline: Dispositivos de campo (Lote 2) que generan eventos de custodia sin conexión almacenan eventos localmente, los sincronizan con microservicio Ledger CoC al reconectar, que los registra en el registro inmutable manteniendo orden temporal correcto.

Tecnología: ImmuDB, instancias en contenedores Docker orquestados por Kubernetes.

Interfaz de entrada: API REST del microservicio Ledger CoC (llamadas desde otros microservicios de THOT o desde Lote 2).

Interfaz de salida: API nativa de ImmuDB (comunicación entre microservicio y el servicio de registro inmutable).

5.1.1.5. Servicio de Sincronización Offline (Offline Sync Service)

Responsabilidad: Gestionar sincronización de datos entre dispositivos de campo (Lote 2) que operan offline y THOT central, garantizando consistencia eventual y resolución de conflictos.

Funcionalidades:

- **Recepción de colas pendientes**: Dispositivo de campo al reconectar envía paquete con todos los eventos generados offline (formato: array de eventos, cada uno con timestamp local, ID dispositivo, checksum). Servicio valida integridad, registra eventos en orden cronológico (según timestamps locales), detecta conflictos.
- **Resolución de conflictos**: Si evento de campo entra en conflicto con dato modificado en central durante desconexión (ej. vestigio modificado en ambos sitios), aplica política configurable:
  - **Last-Writer-Wins (LWW)**: el evento con timestamp más reciente prevalece.
  - **Revisión manual**: marcar conflicto, notificar a usuario para resolución humana.
  - **Política específica por tipo de dato**: ej. LWW para metadatos descriptivos, revisión manual para cambios en resultados de análisis.
- **Priorización de sincronización** (QoS aplicativo): eventos críticos (ej. registro de evidencia con match biométrico) se sincronizan primero, eventos menos urgentes (ej. fotografías de contexto) después. Orden configurable por tipo de evento.
- **Compresión y optimización**: archivos grandes (vídeos, imágenes alta resolución) se comprimen antes de sincronización, se verifica checksum post-compresión.

Tecnología: Microservicio en Python (FastAPI), usa base de datos para tracking de estado de sincronización (PostgreSQL), comunica con dispositivos Lote 2 vía API REST (dispositivo hace POST a endpoint de sincronización con paquete de eventos), comunica con otros servicios de THOT vía bus de eventos Kafka.

Interfaz de entrada: API REST /api/v1/sync/upload (llamada por dispositivos Lote 2 al reconectar).

Interfaz de salida: Publicación de eventos sincronizados en topics de Kafka para procesamiento por otros microservicios.

5.1.1.6. Servicio de Comunicaciones en Tiempo Real (Real-Time Communications Service - OpenVidu)

Responsabilidad: Habilitar streaming de vídeo y audio bidireccional entre expertos en central (usando UI web de THOT) y personal en campo (usando app Lote 2), para asistencia remota en inspecciones.

Arquitectura:

- **OpenVidu Server**: servidor de señalización y gestión de sesiones WebRTC, gestiona creación de "salas" de videoconferencia, admisión de participantes, negociación de conexiones peer-to-peer o mediadas por servidor.
- **Kurento Media Server (KMS)**: servidor de medios que procesa flujos de vídeo/audio (transcodificación, grabación, composición). Usado cuando conexión peer-to-peer directa no es posible (ej. NAT simétrico, firewall estricto).
- **STUN/TURN servers**: servidores auxiliares para traversal de NAT (STUN descubre IP pública del cliente, TURN retransmite tráfico cuando NAT simétrico impide peer-to-peer).

Flujo típico de uso:

1. Experto en central solicita iniciar sesión de asistencia remota a agente en campo (desde UI THOT, botón "Iniciar asistencia remota").
2. THOT crea sesión en OpenVidu Server (POST a API de OpenVidu con IDs de participantes).
3. OpenVidu genera tokens de sesión (uno para experto, uno para agente), los devuelve a THOT.
4. THOT envía token a dispositivo de campo vía MQTT o notificación push.
5. Ambos participantes se conectan a OpenVidu Server con sus tokens (usando SDK cliente OpenVidu en JavaScript/Android/iOS).
6. OpenVidu negocia conexión WebRTC (SDP offer/answer, ICE candidates), establece flujos de vídeo/audio.
7. Ambos participantes ven/oyen al otro en tiempo real.
8. (Opcional) OpenVidu graba sesión en servidor, almacena grabación como evidencia (parte de cadena de custodia).

Gestión de calidad de red: OpenVidu adapta automáticamente bitrate y resolución de vídeo según ancho de banda disponible (adaptive bitrate), prioriza audio sobre vídeo si ancho de banda es muy limitado.

Tecnología: OpenVidu CE (Community Edition, open source) o EE (Enterprise, con soporte comercial, a decidir en Fase II), Kurento Media Server, coturn (implementación open source de TURN server).

Interfaz de entrada: API REST de OpenVidu (llamada por microservicios de THOT), WebSockets para señalización WebRTC (conectan clientes web/móvil).

Interfaz de salida: Flujos RTP (Real-time Transport Protocol) de vídeo/audio entre clientes y KMS.

Almacenamiento de grabaciones: Archivos de vídeo (.mp4) almacenados en object storage (ej. MinIO compatible S3), metadatos de grabación (ID sesión, participantes, timestamp inicio/fin) registrados en BBDD y en cadena de custodia.

5.1.2. Diagrama de componentes lógicos

TODO: Esquema

5.1.3. Flujos de datos principales

5.1.3.1. Flujo 1: Registro de vestigio desde campo

1. Agente en campo (app Lote 2) captura vestigio (fotografía + metadatos).
2. App valida datos localmente contra JSON Schema.
3. Si tiene conexión: POST a https://thot.policia.es/api/v1/vestigios (vía API Gateway) con JSON + archivo adjunto.
4. Si NO tiene conexión: almacena en cola local persistente (SQLite cifrada).
5. API Gateway autentica (valida certificado mTLS del dispositivo), autoriza (verifica que dispositivo tiene rol "campo").
6. API Gateway enruta a microservicio GestionVestigiosService.
7. GestionVestigiosService valida JSON contra esquema, genera ID único para vestigio (UUID).
8. Almacena metadatos en PostgreSQL, archivo en MinIO (object storage).
9. Publica evento vestigio.creado en Kafka topic.
10. CadenaCustodiaService (suscrito a topic) recibe evento, registra en registro inmutable (ImmuDB) (valida y añade evento al registro).
11. ServicioIA (suscrito a topic) recibe evento, si vestigio contiene imagen facial, dispara análisis biométrico automático (invoca adaptador ABIS).
12. Respuesta HTTP 201 Created retornada a app Lote 2 con ID de vestigio.

Tiempo total esperado: <2 segundos con conexión 4G/5G normal.

5.1.3.2. Flujo 2: Consulta biométrica a ABIS desde central

1. Analista en central (UI web THOT) solicita cotejo de huella dactilar de vestigio contra ABIS.
2. UI envía GET a https://thot.policia.es/api/v1/vestigios/{id}/biometria/cotejar.
3. API Gateway autentica (token JWT del analista), autoriza (verifica rol "Analista").
4. Enruta a ServicioBiometria.
5. ServicioBiometria consulta si resultado ya está en caché (Redis), si sí, retorna inmediatamente.
6. Si no está en caché, invoca AdaptadorABIS (microservicio especializado).
7. AdaptadorABIS transforma petición de THOT (JSON) a formato específico de ABIS (puede ser SOAP, protocolo propietario), envía petición a ABIS.
8. ABIS procesa cotejo 1:N (puede tardar 5-10 segundos), retorna lista de candidatos con scores.
9. AdaptadorABIS normaliza respuesta a JSON canónico THOT, retorna a ServicioBiometria.
10. ServicioBiometria almacena resultado en caché (Redis, TTL 1 hora), publica evento analisis.biometrico.completado en Kafka.
11. Retorna resultado a UI (HTTP 200 + JSON con candidatos).

Cofinanciado por la Unión Europea Fondos Europeos

12. UI muestra resultados a analista con visualización (scores, imágenes de candidatos).

Tiempo total esperado: 10-15 segundos primera vez (cotejo real), <1 segundo consultas subsiguientes (caché).

5.1.3.3. Flujo 3: Sincronización offline tras reconexión

1. Dispositivo Lote 2 estuvo 6 horas sin conexión, acumuló 120 eventos en cola local.
2. Al detectar reconexión (listener de conectividad en app), inicia proceso de sincronización.
3. App prepara paquete: array de eventos JSON + checksums + archivos adjuntos, comprime (gzip).
4. POST a https://thot.policia.es/api/v1/sync/upload con paquete comprimido.
5. API Gateway enruta a ServicioSincronizacion.
6. ServicioSincronizacion descomprime paquete, valida checksums, ordena eventos por timestamp.
7. Para cada evento: verifica si ya existe en THOT (por ID evento), si no existe, lo procesa:
   - Almacena metadatos en BBDD.
   - Almacena archivos en MinIO.
   - Publica evento en Kafka (como si acabara de ocurrir, pero con flag sincronizado_offline=true y timestamp_real=<timestamp campo>).
   - Registra en cadena de custodia con timestamp correcto.
8. Si detecta conflicto (evento modificó dato que también se modificó en central), aplica política:
   - Ejemplo: metadata de vestigio modificada en campo y en central → Last-Writer-Wins según timestamp más reciente.
   - Si conflicto no resolvible automáticamente, crea ticket en sistema de gestión de incidencias, notifica a responsable.
9. Retorna respuesta a app: HTTP 200 + resumen (120 eventos sincronizados, 0 conflictos, 0 errores).
10. App elimina eventos de cola local tras confirmación exitosa.

Tiempo total esperado: 2-5 minutos para 120 eventos (depende de tamaño de archivos adjuntos).

5.1.4. Requisitos relacionados de la vista lógica

Componente	Requisitos implementados
API Gateway	RS-INTEROP-01, RS-INTEROP-02, RS-INTEROP-03, RS-INTEROP-04, RNF-INTEROP-01
Servicio Espacio de Datos	RF-INTEROP-02, RF-INTEROP-03, INTEROP-2, INTEROP-4, INT-GEN-2
Bus de Eventos (Kafka)	HW-L1-5, RNF-INTEROP-03, P6 (resiliencia)
Servicio Cadena de Custodia	RF-INTEROP-04, INS-EJE-6, RN-INTEROP-01, RS-INTEROP-04
Servicio Sincronización Offline	RF-INTEROP-01, RNF-INTEROP-04, INTEROP-7

Cofinanciado por la Unión Europea · Fondos Europeos

Componente	Requisitos implementados
Servicio Comunicaciones Tiempo Real	RF-INTEROP-05, INS-EJE-5, INT-16

***Tabla 8. Requisitos relacionados de la vista lógica**. Enumera los componentes principales y los requisitos implementados asociados a cada uno.*

5.2. Vista física / de despliegue

(on-prem, cloud, híbrida)
La vista física describe cómo los componentes lógicos se despliegan en infraestructura real (servidores, contenedores, redes), incluyendo aspectos de escalabilidad, alta disponibilidad y seguridad de red.

5.2.1. Arquitectura de despliegue objetivo

La arquitectura de THOT se despliega sobre **Kubernetes** (orquestador de contenedores), que puede ejecutarse en:

- **On-premise**: servidores físicos en centros de datos de la Policía Nacional (opción preferida inicialmente por ENS Alto).
- **Cloud privado**: OpenStack o similar gestionado por Policía Nacional.
- **Cloud híbrido** (futuro): combinación de on-premise + cloud público para cargas no sensibles.

Decisión de infraestructura física final: A tomar en arranque de Fase II tras análisis conjunto con Policía Nacional, considerando restricciones ENS Alto, costes, capacidad operativa interna.

5.2.2. Cluster Kubernetes de THOT

Configuración mínima:

- **Nodos master (control plane)**: 3 nodos dedicados (tolerancia a fallo de 1 nodo).
  - CPU: 4 cores, RAM: 16 GB, Disco: 100 GB SSD por nodo.
  - Ejecutan: API Server, Scheduler, Controller Manager, etcd (base de datos distribuida de Kubernetes).
- **Nodos worker (carga de trabajo)**: Mínimo 5 nodos inicialmente, escalable a 20+ según carga.
  - CPU: 16 cores, RAM: 64 GB, Disco: 500 GB SSD por nodo.
  - Ejecutan: pods de microservicios, bases de datos, servicios auxiliares.
- **Nodos GPU** (opcional, para cargas de IA intensivas): 2 nodos con GPUs NVIDIA A30 o superior.
  - Para entrenamiento/inferencia de modelos de IA que lo requieran.

Red:

- **Red de gestión** (management): VLAN aislada para administración de cluster (acceso kubectl, dashboards).
- **Red de servicios** (services): VLAN para comunicación entre pods dentro del cluster.
- **Red externa** (external): VLAN para comunicación con internet/exterior (tráfico de API Gateway).

- **Segmentación adicional**: pods con datos sensibles en VLAN separada con firewall que solo permite tráfico desde pods autorizados.

Storage:

- **Almacenamiento persistente** (PersistentVolumes): NFS o iSCSI SAN para bases de datos.
- **Object storage**: MinIO desplegado en cluster o servicio externo compatible S3 (ej. Ceph RGW).

5.2.3. Distribución de componentes en cluster

Componente	Tipo de despliegue	Réplicas mínimas	Recursos por réplica
API Gateway (Kong)	Deployment + Service (LoadBalancer o NodePort)	3	2 CPU, 4 GB RAM
Microservicios de negocio (GestionVestigios, CadenaCustodia, etc.)	Deployment + Service (ClusterIP)	2-5 según carga	1-2 CPU, 2-4 GB RAM
Apache Kafka (brokers)	StatefulSet + Headless Service	3	4 CPU, 8 GB RAM, 500 GB disco
PostgreSQL (principal)	StatefulSet + PersistentVolume	1 master + 2 replicas (streaming replication)	4 CPU, 16 GB RAM, 1 TB disco
MongoDB	StatefulSet (ReplicaSet)	3	2 CPU, 8 GB RAM, 500 GB disco
Redis Cluster	StatefulSet	6 (3 masters + 3 slaves)	2 CPU, 4 GB RAM
Hyperledger Fabric peers	StatefulSet	3	2 CPU, 4 GB RAM, 200 GB disco
Hyperledger Fabric orderers	StatefulSet	3	2 CPU, 4 GB RAM, 100 GB disco

Componente	Tipo de despliegue	Réplicas mínimas	Recursos por réplica
OpenVidu Server	Deployment	2	4 CPU, 8 GB RAM
Kurento Media Server	Deployment (puede escalar según sesiones)	2-10	4 CPU, 8 GB RAM (intensivo CPU)
ElasticSearch (logs)	StatefulSet	3	4 CPU, 16 GB RAM, 1 TB disco
Prometheus (métricas)	StatefulSet	2	2 CPU, 8 GB RAM, 500 GB disco
Grafana (visualización)	Deployment	2	1 CPU, 2 GB RAM

Tabla 9. Distribución de componentes en cluster de THOT. *Enumera para cada componente el tipo de despliegue, el número de réplicas mínimas y los recursos por réplica.*

Nota: Recursos son estimaciones iniciales, se ajustarán en Fase II tras pruebas de carga y monitorización de producción. Kubernetes HPA escalará automáticamente réplicas de Deployments según métricas.

5.2.4. Despliegue de Hyperledger Fabric en Kubernetes

Hyperledger Fabric no está diseñado nativamente para Kubernetes, requiere adaptación:

Opción 1 (recomendada): Usar **Hyperledger Fabric Operator** para Kubernetes, proyecto open source que automatiza despliegue y gestión de redes Fabric en Kubernetes. Despliega peers, orderers, CA como Custom Resources de Kubernetes.

Opción 2: Desplegar manualmente con Helm charts (más control, más complejidad operativa).

Configuración de red Fabric:

- **Organización**: "PoliciaCientifica" (única organización inicialmente, puede añadirse "LaboratoriosExternos" en futuro si se integran laboratorios colaboradores).
- **Peers**: 3 peers de PoliciaCientifica, distribuidos en diferentes nodos Kubernetes para tolerancia a fallos.
- **Orderers**: 3 orderers con algoritmo Raft (CFT - Crash Fault Tolerant, suficiente para entorno confiable).
- **CA (Certificate Authority)**: Fabric CA para emitir certificados de identidad de participantes.

- **Chaincodes**: desplegados como contenedores Docker manejados por peers, versionados, actualizables.

5.2.5. Conectividad con sistemas externos

Lote 2 (dispositivos de campo):

- Conexión vía **internet público** (4G/5G), entrada por **balanceador de carga HTTPS** (puerto 443) delante de API Gateway.
- Autenticación mTLS: cada dispositivo tiene certificado cliente único, emitido por CA de THOT, instalado en almacenamiento seguro del dispositivo (Keystore Android, Keychain iOS).
- Restricción IP (opcional): whitelist de rangos IP de operadores móviles usados por Policía Nacional.

Sistemas TIC internos (PERSONAS, ABIS, etc.):

- Conexión vía **red corporativa interna** de Policía Nacional (VLAN dedicada), sin pasar por internet.
- Comunicación desde microservicios adaptadores en Kubernetes hacia endpoints internos (IPs/DNS internos).
- Firewall perimetral de Kubernetes permite tráfico saliente hacia IPs específicas de sistemas internos, prohíbe tráfico saliente no autorizado.

Sistemas externos internacionales (INTERPOL, EUROPOL, etc.):

- Conexión vía **internet con VPN** o **líneas dedicadas** según acuerdos internacionales.
- Autenticación según protocolo de cada sistema (puede ser mTLS, VPN con certificados, credenciales específicas).

5.2.6. Alta disponibilidad y recuperación ante desastres

Alta disponibilidad (HA):

- **Componentes stateless** (API Gateway, microservicios): múltiples réplicas en diferentes nodos, Kubernetes reinicia automáticamente pods que fallan (liveness probes), balanceo de carga entre réplicas.
- **Componentes stateful** (bases de datos):
  - PostgreSQL: master-slave replication, failover automático con Patroni o similar.
  - MongoDB: ReplicaSet con 3 miembros, automatic failover.
  - Kafka: replicación de particiones en 3 brokers.
  - Redis: master-slave per shard, Redis Sentinel para failover.

Recuperación ante desastres (DR):

- **Backups automáticos**:
  - Bases de datos: snapshots diarios, retención 30 días, almacenados en ubicación externa (otro data center o cloud).
  - o Registro inmutable (ImmuDB): snapshots/backup semanales.
  - Object storage (MinIO): replicación a segundo cluster MinIO en ubicación remota (si infraestructura lo permite).

- **RTO (Recovery Time Objective)**: <4 horas para restaurar servicio básico tras desastre completo.
- **RPO (Recovery Point Objective)**: <24 horas (pérdida máxima de datos: últimas 24h, aunque con backups diarios normalmente <1h).
- **Plan de DR**: Documentado en runbook, incluye: (1) activar cluster de backup en ubicación secundaria, (2) restaurar último backup de BBDD, (3) redirigir tráfico (DNS) a cluster secundario, (4) notificar usuarios de posible pérdida de datos recientes.

5.2.7. Seguridad de red

Firewall de perímetro:

- **Entrada**: Solo puerto 443 (HTTPS) abierto hacia API Gateway, todo otro tráfico entrante bloqueado.
- **Salida**: Tráfico saliente desde Kubernetes hacia internet restringido a IPs/dominios autorizados (actualizaciones de software, sistemas externos), logged.

Segmentación interna (Network Policies de Kubernetes):

- **Política default**: deny all (ningún pod puede comunicarse con otro por defecto).
- **Políticas específicas**: solo pods de microservicios de negocio pueden conectar a pods de bases de datos, solo API Gateway puede recibir tráfico externo, etc.
- **Ejemplo**: pod de GestionVestigiosService puede conectar a PostgreSQL y Kafka, pero NO puede conectar directamente a pod de ImmuDB (debe usar API del ServicioLedgerCoC).

Detección de intrusiones:

- **IDS/IPS** en balanceador de carga (ej. Suricata, Snort), analiza tráfico entrante buscando patrones maliciosos, bloquea/alerta.
- **Monitorización de logs** en tiempo real (ElasticSearch + reglas de detección), alertas ante actividad sospechosa (ej. múltiples intentos de autenticación fallidos, peticiones con payloads extraños).

5.2.8. Diagrama de despliegue físico

TODO: Esquema

5.2.9. Consideraciones de despliegue territorial

Modelo centralizado (opción inicial): Una única instancia de THOT desplegada en centro de datos central de Policía Nacional, todas las unidades territoriales acceden remotamente.

Ventajas: Simplicidad operativa (un solo cluster que mantener), consistencia de datos (único repositorio), menores costes de infraestructura.

Desventajas: Latencia para unidades territoriales distantes (si red corporativa tiene latencias altas), dependencia de conectividad WAN (si falla enlace de unidad territorial, no pueden acceder a THOT).

Modelo federado (opción futura si necesario): Despliegue de nodos "edge" de THOT en unidades territoriales críticas (ej. Madrid, Barcelona, Valencia), con sincronización bidireccional al nodo central.

Ventajas: Menor latencia para operaciones locales, mayor resiliencia (unidad territorial puede seguir operando aunque falle WAN).

Desventajas: Mayor complejidad operativa (múltiples clusters), desafíos de consistencia de datos (sincronización, resolución de conflictos).

Decisión: Iniciar con modelo centralizado en Fase II, evaluar necesidad de federación en Fase III tras análisis de rendimiento y feedback operativo.

5.2.10. Requisitos relacionados de la vista física

Aspecto de despliegue	Requisitos implementados
Kubernetes sobre infraestructura flexible	RP-INTEROP-01, HW-L1-3, HW-L1-4
Alta disponibilidad (réplicas, failover)	RNF-INTEROP-02, HW-L1-7, P6 (resiliencia)
Escalabilidad horizontal (HPA)	RNF-INTEROP-03, HW-L1-5, P5 (escalabilidad)
Seguridad de red (firewall, segmentación)	RS-INTEROP-01, RS-INTEROP-02, SEC-L1-3
Backups y DR	HW-L1-7, principio de disponibilidad

***Tabla 10. Trazabilidad de requisitos de despliegue**. Enumera los aspectos de despliegue y los requisitos implementados asociados a cada uno.*

5.3. Vista de datos

(modelos, dominios de datos, catálogos / ontologías)
La vista de datos describe cómo fluye y se transforma la información a través del sistema de interoperabilidad, desde captura en campo hasta almacenamiento persistente y análisis, incluyendo aspectos de normalización semántica y trazabilidad.

5.3.1. Modelo de flujos de información

5.3.1.1. Flujo de evidencias digitales (campo → central)

TODO: Esquema

Transformaciones de datos en el flujo:

1. **Captura (campo)**: Datos en formato nativo (JPEG, MP4, JSON específico de app).
2. **Normalización (campo)**: Conversión a esquema canónico THOT (JSON según F1.3.1), adición de metadatos técnicos (checksums, timestamps UTC ISO 8601).

Cofinanciado por la Unión Europea
Fondos Europeos

3. **Cifrado (campo)**: Payload cifrado simétricamente, clave de sesión cifrada asimétricamente.
4. **Transmisión**: Datos cifrados en tránsito (TLS 1.3 adicional).
5. **Descifrado y validación (central)**: THOT descifra, valida esquema, enriquece con metadatos internos (ID interno, timestamp recepción).
6. **Almacenamiento multi-capa**:
    - **PostgreSQL**: metadatos relacionales (queryables, indexados).
    - **MinIO**: archivos binarios (inmutables, versionados).
    - **ElasticSearch**: índice de texto completo para búsqueda rápida.
    - **Hyperledger Fabric**: registro de evento de custodia (inmutable, auditable).

Latencia objetivo del flujo end-to-end: <3 segundos (desde POST en campo hasta confirmación HTTP 201 + registro en registro inmutable).

5.3.1.2. Flujo de consultas biométricas (central → ABIS → central)

TODO: Esquema

ransformaciones de datos clave:

1. **Normalización de entrada**: Imagen de huella en formato JPEG/PNG → conversión a WSQ (formato estándar forense de compresión de huellas) + extracción de minucias (puntos característicos).
2. **Adaptación de protocolo**: JSON REST (THOT) ↔ SOAP XML o protocolo propietario (ABIS).
3. **Normalización de salida**: Respuesta heterogénea de ABIS (puede incluir campos propietarios, codificaciones específicas) → JSON canónico THOT con campos estándar (candidatos[], score, metadata).
4. **Enriquecimiento**: Resultado biométrico se enriquece con contexto de THOT (ej. si candidato tiene datos en PERSONAS, se añade filiación completa al resultado).

5.3.2. Normalización semántica de datos

Para garantizar interoperabilidad entre sistemas heterogéneos, THOT aplica **normalización semántica** a datos intercambiados:

Problema: Diferentes sistemas llaman a los mismos conceptos con nombres diferentes (ej. "NIF" vs "DNI" vs "documento_identidad_numero"), usan formatos diferentes (ej. fecha "DD/MM/YYYY" vs "YYYY-MM-DD" vs timestamp UNIX), codifican valores con diferentes taxonomías (ej. tipo de evidencia "huella_dactilar" vs "fingerprint" vs código numérico "23").

Solución: Definir **vocabulario controlado canónico** para THOT, basado en estándares cuando existan:

5.3.2.1. Nomenclatura de campos

Regla: **camelCase** para nombres de campos JSON (consistente con F1.3.1).

Ejemplos:

- vestigioId (no vestigio_id, no id_vestigio)
- fechaHoraCaptura (no fecha_captura, no capture_date)
- ubicacionGPS con subobjetos latitud, longitud (no gps_lat, gps_lon)

5.3.2.2. Tipos de datos y formatos

Concepto	Formato canónico THOT	Estándar	Ejemplo
Timestamp	String ISO 8601 UTC	ISO 8601	"2026-01-21T14:30:00Z"
Fecha (sin hora)	String YYYY-MM-DD	ISO 8601	"2026-01-21"
Coordenadas GPS	Object {latitud: number, longitud: number} en grados decimales WGS84	WGS84	{"latitud": 40.4168, "longitud": -3.7038}
Identificador de persona (España)	String NIF sin guiones/espacios	-	"12345678A"
Hash criptográfico	String hexadecimal SHA-256 (64 caracteres)	SHA-256	"a1b2c3d4..."
Duración	String ISO 8601 duration	ISO 8601	"PT2H30M" (2 horas 30 min)
Tamaño archivo	Number (bytes)	-	1048576 (1 MB)

Tabla 11. Tipos de datos y formatos. *Enumera los conceptos de datos y su representación en el formato canónico de THOT, indicando el estándar aplicable y un ejemplo.*

5.3.2.3. Codificaciones de valores

Para campos con valores cerrados (enumeraciones), definir **listas controladas** en JSON Schema:

Ejemplo: Tipo de vestigio

```
{
  "tipoVestigio": {
    "type": "string",
```


Cofinanciado por la Unión Europea
Fondos Europeos

```
    "enum": [
      "huella_dactilar",
      "huella_palmar",
      "adn_biologico",
      "fibra_textil",
      "proyectil_balistica",
      "documento_papel",
      "dispositivo_electronico",
      "sustancia_quimica",
      "imagen_fotografica",
      "video_grabacion",
      "audio_grabacion",
      "otro"
    ]
  }
}
```

Mapeo desde sistemas externos: Cada adaptador mantiene tabla de mapeo desde codificación externa a codificación canónica THOT.

Ejemplo (adaptador ABIS):

```
ABIS código "FP" → THOT "huella_dactilar"
ABIS código "PP" → THOT "huella_palmar"
ABIS código "DNA" → THOT "adn_biologico"
```

5.3.3. JSON-LD para enriquecimiento semántico

JSON-LD (JSON for Linking Data) permite añadir contexto semántico a documentos JSON, facilitando interoperabilidad con sistemas que usan ontologías formales (RDF/OWL).

Caso de uso en THOT: Exportación de datos forenses a sistemas internacionales que requieren formatos semánticamente enriquecidos (ej. INTERPOL, EUROPOL pueden usar ontologías específicas para casos internacionales).

Ejemplo: Vestigio en JSON canónico THOT con contexto JSON-LD:

```
{
```

```
"@context": {
  "@vocab": "https://thot.policia.es/ontology/forense#",
  "dc": "http://purl.org/dc/terms/",
  "xsd": "http://www.w3.org/2001/XMLSchema#"
},
"@type": "Vestigio",
"@id": "https://thot.policia.es/vestigios/a1b2c3d4-uuid",
"vestigioId": "a1b2c3d4-uuid",
"tipoVestigio": "huella_dactilar",
"asuntoId": "ASU-2026-00123",
"fechaHoraCaptura": {
  "@type": "xsd:dateTime",
  "@value": "2026-01-21T14:30:00Z"
},
"ubicacionCaptura": {
  "@type": "Ubicacion",
  "direccion": "Calle Gran Vía 123, Madrid",
  "coordenadas": {
    "@type": "geo:Point",
    "latitud": 40.4168,
    "longitud": -3.7038
  }
},
"dc:creator": "Juan Pérez (Técnico Policía Científica)",
"dc:created": "2026-01-21T14:30:00Z"
}
```


Ventajas:

- Sistemas externos pueden entender semántica de campos sin conocer esquema específico de THOT (leen @context).
- Permite enlazar datos de THOT con datasets externos (ej. ontologías geográficas, taxonomías forenses internacionales).
- Facilita consultas federadas (SPARQL sobre múltiples sistemas).

Obligatoriedad: JSON-LD es **opcional** para intercambios internos THOT-Lote 2 (suficiente con JSON Schema), **recomendado** para exportaciones a sistemas internacionales.

5.3.4. Gestión de calidad de datos

Para garantizar fiabilidad de información forense, se aplican controles de calidad en múltiples capas:

5.3.4.1. Validación en origen (dispositivos Lote 2)

- **Validación JSON Schema**: Antes de transmitir, app verifica que datos cumplen esquema (tipos correctos, campos obligatorios presentes, valores dentro de rangos permitidos).
- **Checksums**: Calcular hash SHA-256 de archivos binarios antes de transmisión, incluir en metadatos, verificar tras recepción que coincide (detecta corrupción en tránsito).
- **Timestamps fiables**: Dispositivos de campo sincronizan reloj con servidores NTP confiables, incluyen timestamp en metadatos (crítico para cadena de custodia).

5.3.4.2. Validación en destino (THOT central)

- **Revalidación de esquema**: Aunque campo validó, THOT revalida (defensa en profundidad, protege contra clientes maliciosos o bugs en app campo).
- **Verificación de checksums**: Recalcular hashes, comparar con valores recibidos, rechazar si no coinciden.
- **Detección de anomalías**: Reglas de negocio detectan datos sospechosos (ej. GPS indica ubicación en medio del océano, timestamp en el futuro, tamaño de archivo inconsistente con tipo declarado).

5.3.4.3. Métricas de calidad de datos

Monitorizar indicadores de calidad en tiempo real (dashboard Grafana):

Métrica	Definición	Umbral aceptable	Acción si se supera
Tasa de rechazo por schema	% de peticiones rechazadas por validación JSON Schema	<1%	Alertar a equipo Lote 2 (posible bug en app)
Tasa de error de checksum	% de archivos con checksum incorrecto	<0.1%	Alertar (posible problema de red o seguridad)
Datos faltantes obligatorios	% de registros con campos obligatorios NULL	<0.5%	Revisar validación en origen
Duplicados detectados	% de vestigios con ID duplicado o checksum idéntico a existente	<0.01%	Investigar (posible reenvío no intencionado)

***Tabla 12. Métricas de calidad de datos**. Enumera las métricas, su definición, el umbral aceptable y la acción prevista si se supera.*

Cofinanciado por la Unión Europea
Fondos Europeos

5.3.4.4. Enriquecimiento y corrección de datos

- **Georreferenciación inversa**: Si vestigio tiene coordenadas GPS, THOT consulta servicio de geocodificación (ej. Nominatim OSM) para obtener dirección textual, añade a metadatos.
- **Normalización de NIF/NIE**: Si vestigio referencia persona con NIF, normalizar formato (mayúsculas, quitar guiones), validar checksum de NIF (algoritmo estándar).
- **Detección de idioma en textos**: Si vestigio es documento de texto, detectar idioma automáticamente (librería langdetect), almacenar en metadata (facilita búsquedas posteriores).

5.3.5. Gestión de archivos binarios grandes

Vestigios pueden incluir archivos grandes (vídeos 4K, imágenes alta resolución, volcados forenses de discos). Requieren tratamiento especial:

5.3.5.1. Estrategia de almacenamiento

- **Archivos <10 MB**: Transmitir completos en petición HTTP (base64 en JSON o multipart/form-data).
- **Archivos 10 MB - 1 GB**: Transmitir con **resumable upload** (chunked upload): cliente divide archivo en chunks de 5 MB, envía secuencialmente con reintentos por chunk (si falla chunk 3, reenvía solo ese chunk, no todo el archivo).
- **Archivos >1 GB**: Usar **presigned URLs** de MinIO: cliente solicita a THOT permiso para subir archivo directamente a MinIO (sin pasar por API Gateway), THOT genera URL firmada temporalmente (válida 1 hora), cliente sube directamente a MinIO con esa URL, notifica a THOT al completar.

5.3.5.2. Compresión

- **Imágenes**: No recomprimir JPEGs (introduce pérdida adicional), almacenar originales. Para PNGs grandes, convertir a formato más eficiente (JPEG si tolera pérdida, WebP si cliente soporta).
- **Vídeos**: Si cliente sube vídeos no comprimidos o con códec ineficiente, THOT puede transcodificar a H.265/HEVC (mejor compresión que H.264, mantiene calidad forense) en background task.
- **Documentos**: PDFs grandes comprimir con Ghostscript (downsampling de imágenes internas, eliminar metadatos innecesarios), mantener original como backup.

5.3.5.3. Checksums de integridad

- **SHA-256 de archivo completo**: Calculado en origen, verificado en destino, almacenado en metadatos y en registro inmutable (ImmuDB) (cadena de custodia)
- **Checksums de chunks** (para archivos grandes): Si archivo se transmite en chunks, calcular SHA-256 de cada chunk, THOT verifica chunk by chunk antes de reensamblar, detecta corrupción parcial.

Cofinanciado por la Unión Europea
Fondos Europeos

5.3.6. Requisitos relacionados de la vista de datos

Aspecto de datos	Requisitos implementados
Normalización semántica (vocabularios, JSON-LD)	INTEROP-4, RF-INTEROP-02, RCD-INTEROP-01
Validación y calidad de datos	RCD-INTEROP-02, RS-INTEROP-03
Gestión de archivos grandes	RF-INTEROP-01, HW-L1-3
Trazabilidad de transformaciones	RF-INTEROP-04, RN-INTEROP-01 (ISO 21043)

Tabla 13. Requisitos asociados a la vista de datos.

5.4. Vista de seguridad

(zonas, DMZ, firewalls, segmentación de redes)

La vista de seguridad describe los mecanismos de protección aplicados a las interacciones de interoperabilidad, cubriendo autenticación, autorización, cifrado, auditoría y cumplimiento de ENS Alto.

5.4.1. Arquitectura de seguridad multi-capa

La seguridad de interoperabilidad se implementa en múltiples capas (defensa en profundidad), de forma que compromiso de una capa no comprometa el sistema completo:

TODO: Esquema

5.4.2. Autenticación de actores

Diferentes tipos de actores acceden a THOT, cada uno con mecanismo de autenticación apropiado:

5.4.2.1. Usuarios humanos (analistas, técnicos, investigadores)

Protocolo: OAuth 2.0 + OpenID Connect (OIDC)

Flujo típico (Authorization Code Flow):

1. Usuario accede a UI web de THOT.
2. UI redirige a **Identity Provider (IdP)** de Policía Nacional (ej. Keycloak, Azure AD).
3. Usuario se autentica en IdP (usuario/contraseña + MFA si está habilitado).
4. IdP retorna **authorization code** a UI.
5. UI intercambia code por **access token** (JWT) y **refresh token** (llamando a token endpoint de IdP).
6. UI incluye access token en header Authorization: Bearer <token> de todas las peticiones a API de THOT.

7. API Gateway valida token (verifica firma con clave pública de IdP, verifica no expirado, verifica issuer correcto).
8. Si token válido, extrae userId y roles de claims del JWT, permite petición.
9. Cuando access token expira (TTL típico: 15 minutos), UI usa refresh token para obtener nuevo access token sin requerir reautenticación de usuario.

Claims obligatorios en JWT:

```
{
  "sub": "user123",          // ID único de usuario
  "name": "Juan Pérez",      // Nombre completo
  "email": "juan.perez@policia.es",
  "roles": ["analista", "consulta_biometria"],
  "iss": "https://idp.policia.es",
  "aud": "thot-api",
  "exp": 1737469200,         // Timestamp expiración
  "iat": 1737468300          // Timestamp emisión
}
```

Ventajas:

- **Single Sign-On (SSO)**: Usuario se autentica una vez en IdP, accede a THOT y otros sistemas sin reautenticación.
- **Centralización**: Gestión de usuarios (altas, bajas, cambios de roles) se hace en IdP corporativo, THOT consume.
- **MFA soportado**: Si IdP requiere MFA (SMS, app autenticadora, biometría), aplica transparentemente a THOT.

5.4.2.2. Dispositivos de campo (Lote 2, máquina-a-máquina)

Protocolo: Mutual TLS (mTLS)

Flujo típico:

1. Dispositivo de campo tiene certificado cliente X.509 instalado (firmado por CA de THOT), almacenado en almacenamiento seguro del dispositivo (Android Keystore, iOS Keychain).
2. Al conectar a THOT, dispositivo y servidor establecen handshake TLS.
3. Servidor presenta certificado servidor (validado por dispositivo contra CA raíz de THOT).
4. Servidor solicita certificado cliente (client certificate request).
5. Dispositivo presenta certificado cliente.
6. Servidor valida certificado cliente:
   - ¿Firmado por CA de confianza (CA de THOT)?
   - ¿No revocado (consulta CRL o OCSP)?
   - ¿No expirado?

   - ¿Subject Name o SAN corresponde a dispositivo autorizado (ej. CN=dispositivo-campo-00123)?
7. Si certificado válido, conexión TLS se establece con autenticación mutua.
8. API Gateway extrae deviceId del certificado, verifica permisos asociados a ese dispositivo (consulta base de datos de dispositivos autorizados).

Ventajas:

- **Autenticación fuerte**: Compromiso de contraseña no compromete autenticación (se necesita clave privada de certificado).
- **Sin intercambio de credenciales**: No se envían usuario/contraseña en cada petición (más seguro que Basic Auth).
- **Detección de dispositivos comprometidos**: Si dispositivo es robado/perdido, se revoca su certificado (añadiendo a CRL), inmediatamente pierde acceso.

Gestión de certificados:

- **Emisión**: Al aprovisionar nuevo dispositivo, THOT CA genera certificado (TTL: 1 año), se instala en dispositivo mediante proceso seguro (ej. durante configuración inicial en comisaría, con autenticación del técnico).
- **Renovación**: 30 días antes de expiración, dispositivo solicita renovación automática (presentando certificado actual aún válido), recibe nuevo certificado.
- **Revocación**: Si dispositivo se pierde, técnico lo marca como revocado en interfaz de gestión de THOT, certificado se añade a CRL, dispositivo no puede autenticarse.

5.4.2.3. Sistemas externos (ABIS, PERSONAS, EUROPOL, etc.)

Protocolo: Depende del sistema (mTLS preferido, OAuth 2.0 Client Credentials si el sistema lo soporta, credenciales API Key como fallback).

Caso preferido: mTLS:

- THOT actúa como cliente, presenta certificado cliente a sistema externo.
- Similar a flujo de dispositivos Lote 2, pero en dirección inversa.

Caso OAuth 2.0 Client Credentials:

- THOT (actuando como cliente OAuth) se autentica contra token endpoint del sistema externo con client_id + client_secret.
- Recibe access token, lo incluye en peticiones posteriores.

Caso API Key (menos seguro, solo si sistema externo no soporta mTLS/OAuth):

- API Key almacenada cifrada en HashiCorp Vault de THOT.
- Microservicio adaptador recupera API Key de Vault en tiempo de ejecución, la incluye en header (X-API-Key) de peticiones a sistema externo.
- API Keys se rotan periódicamente (cada 90 días, proceso manual coordinado con administrador del sistema externo).

5.4.3. Autorización granular (RBAC)

Una vez autenticado, se aplica **control de acceso basado en roles (RBAC)** para determinar qué operaciones puede realizar cada actor.

5.4.3.1. Roles definidos

Rol	Descripción	Permisos típicos
admin	Administrador del sistema	Todos los permisos + gestión de usuarios/roles
analista	Analista forense (central)	Consultar vestigios, solicitar análisis, ver resultados, NO modificar evidencias originales
tecnico_campo	Técnico Policía Científica (campo)	Registrar vestigios, actualizar CoC, subir archivos, consultar sus propios casos
investigador	Investigador de caso (no científico)	Consultar vestigios de casos asignados, ver resultados, NO solicitar análisis directamente
auditor	Auditor interno/externo	Solo lectura completa de logs y cadena de custodia, NO acceso a datos operativos
sistema_externo	Sistema automatizado (Lote 2, integración con otro sistema)	Registrar vestigios, consultar APIs específicas, NO acceso a funciones de gestión

Tabla 14. Roles RBAC y permisos típicos.

5.4.3.2. Matriz de permisos (ejemplo simplificado)

Operación (API endpoint)	admin	analista	tecnico_campo	investigador	auditor	sistema_externo
GET /api/v1/vestigios/{id}	✓	✓	✓ (solo sus casos)	✓ (solo sus casos)	✗	✓ (con restricción)
POST /api/v1/vestigios	✓	✗	✓	✗	✗	✓

Cofinanciado por la Unión Europea
Fondos Europeos

Operación (API endpoint)	admin	analista	tecnico_campo	investigador	auditor	sistema_externo
PUT /api/v1/vestigios/{id}	✓	✗	✓ (solo sus vestigios)	✗	✗	✗
POST /api/v1/analisis/biometria	✓	✓	✗	✗	✗	✗
GET /api/v1/coc/{id}/historial	✓	✓	✓	✓ (solo sus casos)	✓	✗
GET /api/v1/admin/usuarios	✓	✗	✗	✗	✗	✗
GET /api/v1/logs/auditoria	✓	✗	✗	✗	✓	✗

Tabla 15. Ejemplo simplificado de matriz RBAC para acceso a endpoints críticos.

5.4.3.3. Implementación de autorización

En API Gateway (primera línea):

- Valida token, extrae roles.
- Aplica políticas básicas (ej. "endpoint /admin/* solo accesible por rol admin").
- Si viola política, retorna HTTP 403 Forbidden inmediatamente sin llamar a backend.

En microservicios (defensa en profundidad):

- Revalidan roles (no confían ciegamente en API Gateway).
- Aplican **autorización basada en recursos**: Ej. técnico de campo solicita modificar vestigio, microservicio verifica que el vestigio fue registrado por ese técnico o por su unidad (consulta BBDD), si no, retorna 403 aunque el rol en general permita modificar vestigios.

Cofinanciado por la Unión Europea
Fondos Europeos

5.4.4. Cifrado de datos

5.4.4.1. Datos en tránsito

TLS 1.3 obligatorio para todas las conexiones:

- **Clientes externos → API Gateway**: TLS 1.3 con cipher suites fuertes (ej. TLS_AES_256_GCM_SHA384), certificado servidor emitido por CA reconocida (ej. Let's Encrypt para entornos internet-facing, CA corporativa para entornos internos).
- **API Gateway → Microservicios** (tráfico interno cluster): Puede ser HTTP plano si cluster está en red aislada confiable (más rendimiento), o mTLS si política de seguridad lo exige (paranoia justificada para ENS Alto).
- **Microservicios → Sistemas externos**: TLS 1.3 (o TLS 1.2 como mínimo si sistema externo no soporta 1.3).

Configuración TLS reforzada:

- Deshabilitar TLS 1.0 y 1.1 (vulnerables).
- Deshabilitar cipher suites débiles (ej. RC4, DES, MD5).
- Habilitar Perfect Forward Secrecy (PFS) con cipher suites ECDHE.
- Configurar HSTS (HTTP Strict Transport Security): header Strict-Transport-Security: max-age=31536000; includeSubDomains para forzar uso de HTTPS en clientes.

5.4.4.2. Datos en reposo

Cifrado de bases de datos:

- **PostgreSQL**: Transparent Data Encryption (TDE) a nivel de filesystem (LUKS en Linux) o a nivel de PostgreSQL (extensión pgcrypto para cifrado de columnas sensibles específicas).
- **MongoDB**: Encryption at Rest habilitado (AES-256, claves gestionadas por KMIP-compatible key manager).
- **Redis**: Aunque datos en Redis son típicamente efímeros (caché), cifrar disco si Redis persiste datos (RDB snapshots) con dm-crypt.

Cifrado de object storage (MinIO):

- **Server-Side Encryption (SSE)**: MinIO cifra objetos al escribirlos al disco (AES-256), descifra al leerlos, transparente para clientes.
- **Claves de cifrado**: Gestionadas por MinIO KMS (integrado con HashiCorp Vault), una clave maestra (master key) cifra claves de datos (data keys) usadas para cada objeto.

Gestión de claves criptográficas (HashiCorp Vault):

- **Almacenamiento seguro de secretos**: Claves de cifrado, API keys, credenciales de BBDD, certificados privados, almacenados en Vault cifrados en reposo.

- **Control de acceso granular**: Cada microservicio solo puede leer secretos necesarios para su función (ej. microservicio GestionVestigios puede leer clave de cifrado de BBDD, pero NO puede leer API Key de ABIS).
- **Rotación automática de claves**: Vault puede rotar claves automáticamente (ej. cada 90 días), reencripta datos con nueva clave, mantiene versiones antiguas para descifrar datos históricos.
- **Auditoría de acceso a secretos**: Vault registra todo acceso a secretos (qué microservicio, qué secreto, cuándo), logs inmutables.

5.4.5. Auditoría y cumplimiento

5.4.5.1. Logging de eventos de auditoría

Todas las operaciones de interoperabilidad generan **logs de auditoría estructurados** con información obligatoria:

Campos obligatorios en log de auditoría:

```
{
  "timestamp": "2026-01-21T14:30:00.123Z",  // UTC, precisión milisegundos
  "eventType": "vestigio.creado",           // Tipo de evento
  "actorType": "usuario",                   // "usuario", "dispositivo", "sistema"
  "actorId": "user123",                     // ID de quien realizó operación
  "actorName": "Juan Pérez",                // Nombre legible
  "actorIP": "192.168.1.100",               // IP origen
  "resource": "vestigio",                   // Recurso afectado
  "resourceId": "a1b2c3d4-uuid",            // ID específico del recurso
  "action": "create",                       // Acción realizada
  "result": "success",                      // "success", "failure"
  "details": {                              // Detalles específicos del evento
    "asuntoId": "ASU-2026-00123",
    "tipoVestigio": "huella_dactilar"
  },
  "sessionId": "sess-xyz789",               // ID de sesión (para correlacionar eventos)
  "traceId": "trace-abc123"                 // ID de traza distribuida
}
```

Destino de logs:

- **ElasticSearch**: Almacenamiento central de logs, indexado, queryable, visualizable en Kibana.

- **Object Storage (MinIO) como backup inmutable**: Logs se exportan diariamente a object storage en formato JSON comprimido, con WORM (Write Once Read Many) habilitado para inmutabilidad (cumplimiento ENS Alto requisito de auditoría).
- **Retención**: Logs de auditoría se retienen mínimo **7 años** (requisito ENS Alto), después se archivan (pueden comprimirse/moverse a almacenamiento barato pero no se eliminan).

5.4.5.2. SIEM integration

Los logs de auditoría se envían a **SIEM (Security Information and Event Management)** de Policía Nacional (si existe), o THOT despliega SIEM propio (ej. Wazuh, open source) para:

- **Correlación de eventos**: Detectar patrones de actividad sospechosa (ej. mismo usuario intenta acceder a 100 vestigios de casos no asignados en 5 minutos → posible fuga de datos).
- **Alertas automáticas**: Disparar alertas ante eventos críticos:
  - Múltiples intentos de autenticación fallidos (posible brute force).
  - Acceso a datos sensibles desde IP no habitual.
  - Modificación de vestigio fuera de horario laboral.
  - Cambio en configuración de seguridad (ej. rol de usuario elevado a admin).
- **Dashboards de seguridad**: Visualización en tiempo real de métricas de seguridad (intentos de autenticación, eventos de acceso, alertas disparadas).

5.4.5.3. Cumplimiento ENS Alto

Controles de seguridad implementados (mapeo a ENS):

Control ENS	Medida implementada en THOT
[op.acc.1] Identificación	mTLS (dispositivos), OAuth 2.0 + OIDC (usuarios)
[op.acc.2] Requisitos de acceso	Autenticación obligatoria, MFA recomendado en IdP
[op.acc.3] Segregación de funciones	RBAC con roles granulares, principio de mínimo privilegio
[op.acc.4] Proceso de gestión de derechos	Gestión de usuarios/roles en IdP corporativo, auditoría de cambios
[op.acc.5] Mecanismo de autenticación	Certificados X.509 (mTLS), JWT firmados (OAuth 2.0)
[op.acc.6] Acceso local	Acceso administrativo a servidores restringido (bastion hosts, 2FA)
[mp.com.1] Perímetro seguro	Firewall perimetral, solo puerto 443 abierto
[mp.com.2] Protección de la confidencialidad	TLS 1.3 obligatorio

Control ENS	Medida implementada en THOT
[mp.com.3] Protección de la integridad	Checksums (SHA-256), firmas digitales en registro inmutable
[mp.info.3] Cifrado	TLS 1.3 (tránsito), AES-256 (reposo)
[op.exp.8] Registro de actividad	Logs de auditoría completos, inmutables, retenidos 7 años
[op.exp.9] Registro de gestión	Logs de operaciones de administración (cambios de config, despliegues)
[op.cont.1] Análisis de riesgos	Análisis de riesgos realizado en Fase I, actualizado anualmente
[op.cont.2] Plan de continuidad	Plan DR documentado (sección 5.2), probado anualmente

Tabla 16. Correspondencia entre controles ENS y medidas técnicas implementadas en THOT para control de acceso, comunicaciones seguras, cifrado, auditoría y continuidad.

Auditorías de cumplimiento:

- **Auditoría interna** (anual): Equipo de seguridad de Policía Nacional revisa cumplimiento de controles ENS.
- **Auditoría externa** (bienal): Entidad certificadora externa (acreditada por ENAC) audita cumplimiento ENS Alto, emite certificado si conforme.

5.4.6. Protección contra amenazas comunes

5.4.6.1. Inyección de código (SQL Injection, NoSQL Injection)

Mitigación:

- **Prepared statements / parametrized queries**: Toda consulta a BBDD usa parámetros (no concatenación de strings).
- **ORMs (Object-Relational Mappers)**: Uso de frameworks (ej. Hibernate para Java, SQLAlchemy para Python) que generan queries seguras.
- **Validación de entrada**: JSON Schema rechaza payloads con caracteres sospechosos antes de llegar a capa de persistencia.

5.4.6.2. Cross-Site Scripting (XSS)

Mitigación (aunque THOT es principalmente API backend, si hay componentes web):

- **Output encoding**: Todo dato dinámico insertado en HTML se escapa (ej. convertir **<** a <).

- **Content Security Policy (CSP)**: Header HTTP Content-Security-Policy restringe fuentes de scripts permitidas.

5.4.6.3. Cross-Site Request Forgery (CSRF)

Mitigación:

- **CSRF tokens**: Peticiones que modifican estado (POST/PUT/DELETE) requieren token anti-CSRF generado por servidor, incluido en petición.
- **SameSite cookies**: Cookies de sesión (si se usan) con flag SameSite=Strict.

5.4.6.4. Denial of Service (DoS / DDoS)

Mitigación:

- **Rate limiting en API Gateway**: Límite de peticiones por IP/usuario (ej. 1000 peticiones/hora por usuario).
- **CAPTCHA en endpoints públicos** (si aplicable).
- **DDoS mitigation en balanceador de carga**: Detección de patrones de tráfico anómalo (ej. Cloudflare, AWS Shield, o solución on-premise).
- **Escalado automático**: Kubernetes HPA escala pods ante incremento de carga (aunque tiene límite, protege contra DoS moderados).

5.4.6.5. Man-in-the-Middle (MitM)

Mitigación:

- **mTLS obligatorio**: Autentica ambos extremos de la comunicación, previene MitM.
- **Certificate pinning en apps móviles** (Lote 2): App de campo tiene certificado servidor de THOT "pinned" (hardcoded hash del certificado), rechaza cualquier certificado diferente aunque esté firmado por CA de confianza (protege contra CA comprometida o certificado falso).

5.4.7. Requisitos relacionados de la vista de seguridad

Mecanismo de seguridad	Requisitos implementados
Autenticación (mTLS, OAuth 2.0)	RS-INTEROP-01, SEC-L1-3
Autorización (RBAC)	RS-INTEROP-03, SEC-L1-3
Cifrado (TLS 1.3, AES-256)	RS-INTEROP-02, SEC-L1-3, ENS Alto [mp.info.3]
Auditoría (logs inmutables)	RS-INTEROP-04, ENS Alto [op.exp.8], RN-INTEROP-01 (ISO 21043)

Mecanismo de seguridad	Requisitos implementados
Gestión de claves (Vault)	RS-INTEROP-02, ENS Alto [mp.info.3]

Tabla 17. Mapeo de mecanismos de seguridad en THOT a requisitos de interoperabilidad y seguridad (RS/SEC) y a controles ENS aplicables (cifrado, auditoría y gestión de claves).

5.5. Vista de interoperabilidad Lote 1 y Lote 2

Flujos en tiempo real y de asistencia remota entre lotes

Esta vista se enfoca específicamente en la arquitectura de interoperabilidad entre THOT (Lote 1) y las soluciones tecnológicas de campo (Lote 2), detallando protocolos, flujos de datos bidireccionales y sincronización.

5.5.1. Arquitectura de comunicación THOT – Lote 2

TODO: Esquema

5.5.2. Protocolos de comunicación por caso de uso

5.5.2.1. Caso 1: Operaciones CRUD síncronas (REST)

Casos de uso: Registro de vestigio, consulta de asunto, actualización de datos.

Protocolo: HTTP REST over TLS 1.3 + mTLS

Endpoints principales (definidos en F1.3.1):

Método	Endpoint	Descripción	Request Body	Response	Requiere conexión
POST	/api/v1/vestigios	Registrar nuevo vestigio	JSON vestigio + archivo (multipart)	201 + {vestigioId}	Sí (fallback offline)
GET	/api/v1/vestigios/{id}	Consultar vestigio	-	200 + JSON vestigio	Sí (caché local posible)
PATCH	/api/v1/vestigios/{id}	Actualizar parcialmente vestigio	JSON con campos a modificar	200 + JSON vestigio actualizado	Sí (fallback offline)
POST	/api/v1/asuntos	Crear asunto	JSON asunto	201 + {asuntoId}	Sí

GET	/api/v1/asuntos/{id}	Consultar asunto	-	200 + JSON asunto	Sí (caché local posible)
POST	/api/v1/coc/eventos	Registrar evento de cadena de custodia	JSON evento CoC	201 + {eventoId}	Sí (cola offline crítica)

Tabla 18. Resumen de endpoints REST principales de THOT para gestión de vestigios, asuntos y eventos de Cadena de Custodia, incluyendo método, payload esperado y requisitos de conectividad (con mecanismos de caché/cola offline cuando aplique).

Comportamiento offline (app Lote 2):

- Si dispositivo detecta ausencia de conexión (timeout o error de red), operaciones de escritura (POST/PUT/PATCH) se encolan localmente en SQLite cifrada.
- Operaciones de lectura (GET) retornan datos de caché local si disponible, o error "sin conexión" si no hay caché.
- Al reconectar, cola local se sincroniza automáticamente con endpoint /api/v1/sync/upload.

5.5.2.2. Caso 2: Telemetría asíncrona (MQTT)

Casos de uso: Envío de métricas de dispositivo (geolocalización, batería, temperatura), notificaciones de estado, eventos de bajo nivel.

Protocolo: MQTT v5.0 over TLS 1.3

Broker MQTT: Mosquitto o HiveMQ desplegado en cluster Kubernetes de THOT (o servicio gestionado compatible).

Topics MQTT (estructura jerárquica):

```
thot/
├── campo/
│   ├── dispositivo/{deviceId}/telemetria
│   │   ├── gps            # Coordenadas GPS en tiempo real
│   │   ├── bateria        # Nivel de batería (%)
│   │   ├── red            # Tipo de conexión (4G/5G/WiFi) + calidad señal
│   │   └── sensores       # Datos de sensores especializados (temperatura, humedad...)
│   │
│   ├── dispositivo/{deviceId}/estado
│   │   ├── online         # Dispositivo se conectó
│   │   ├── offline        # Dispositivo se desconectó
│   │   └── operando       # Dispositivo en operación activa (inspección en curso)
│   │
```

```
│   └── dispositivo/{deviceId}/alertas
│       ├── bateria_baja     # Batería <20%
│       └── error            # Error en app (crash, problema hardware)
│
└── central/
    └── notificaciones/{deviceId}
        ├── asignacion_caso  # Nuevo caso asignado a agente de ese dispositivo
        ├── mensaje_urgente  # Mensaje de coordinador central
        └── actualizacion    # Actualización disponible de app
```

Publicación (dispositivo → THOT):

- Dispositivo se conecta a broker MQTT (autenticación mTLS o usuario/contraseña generado automáticamente).
- Publica mensajes JSON en topics apropiados:

```
{
  "topic": "thot/campo/dispositivo/DEV-00123/telemetria/gps",
  "payload": {
    "timestamp": "2026-01-21T14:30:00Z",
    "latitud": 40.4168,
    "longitud": -3.7038,
    "precision": 5.0,
    "velocidad": 0.0
  },
  "qos": 1
}
```

Suscripción (dispositivo escucha THOT):

- Dispositivo se suscribe a topic personal thot/central/notificaciones/{deviceId}/#.
- Recibe notificaciones push de THOT (ej. nuevo caso asignado, mensaje urgente), reacciona localmente (muestra notificación al usuario, actualiza UI).

QoS (Quality of Service) de MQTT:

- **QoS 0** (at most once): Para telemetría no crítica (GPS en movimiento, batería). Si mensaje se pierde, no pasa nada (siguiente mensaje llegará).

- **QoS 1** (at least once): Para eventos importantes (dispositivo online, error). Garantiza entrega, puede haber duplicados (consumidor debe ser idempotente).
- **QoS 2** (exactly once): Para eventos críticos (no usado típicamente por overhead, preferir QoS 1 + idempotencia).

Ventajas de MQTT vs REST para telemetría:

- **Menor latencia**: Conexión persistente (no handshake TLS por cada mensaje).
- **Menor overhead**: Headers MQTT más pequeños que HTTP.
- **Pub/Sub desacoplado**: Múltiples consumidores pueden suscribirse a mismo topic sin modificar emisor.
- **Soporte offline**: MQTT cliente puede bufferear mensajes localmente mientras está desconectado (QoS>0), envía al reconectar.

5.5.2.3. Caso 3: Streaming de vídeo en tiempo real (WebRTC)

Caso de uso: Asistencia remota, experto en central ve en vivo lo que ve cámara de dispositivo de campo, puede dar instrucciones por audio bidireccional.

Protocolo: WebRTC (SRTP para medios, ICE/STUN/TURN para traversal de NAT)

Infraestructura: OpenVidu Server (coordinación) + Kurento Media Server (procesamiento de medios).

Flujo de inicio de sesión (detallado en sección 5.1, Servicio de Comunicaciones Tiempo Real):

1. Experto en central solicita asistencia remota a agente en campo (desde UI THOT, botón con ID del caso).
2. THOT crea sesión OpenVidu, genera tokens para ambos participantes.
3. THOT envía notificación a dispositivo de campo:
    - Vía MQTT (topic thot/central/notificaciones/{deviceId}/asistencia_remota).
    - Vía push notification (FCM para Android, APNs para iOS) si app está en background.
4. Dispositivo muestra alerta a agente: "Solicitud de asistencia remota de [Nombre Experto]. ¿Aceptar?".
5. Si agente acepta:
    - App de campo se conecta a OpenVidu Server con token recibido.
    - Negotia conexión WebRTC (exchange SDP offer/answer, ICE candidates).
    - Establece flujos de medios (vídeo de cámara campo → experto, vídeo cámara experto → campo opcional, audio bidireccional).
6. Experto ve en tiempo real lo que ve cámara de campo, puede pausar captura de pantalla (screenshot) de frame específico si necesita.
7. Al finalizar asistencia, cualquier participante puede colgar, sesión se cierra, grabación se almacena en MinIO como evidencia.

Calidad de vídeo adaptativa:

- OpenVidu/Kurento ajusta bitrate y resolución dinámicamente según ancho de banda disponible (mide RTT, packet loss).
- En conexión 4G buena: 720p @ 30fps.
- En conexión 4G degradada: 360p @ 15fps.

- Si ancho de banda cae críticamente: solo audio (desactiva vídeo temporalmente).

Grabación de sesiones:

- **Obligatorio para cadena de custodia**: Toda sesión de asistencia remota es parte de la inspección oficial, debe grabarse.
- Kurento Media Server graba vídeo compuesto (both participants en un frame) + audio, genera archivo .mp4.
- Metadata de grabación (ID sesión, participantes, timestamp inicio/fin, caso asociado) se registra en registro inmutable (cadena de custodia).
- Archivo se almacena en MinIO, enlazado al vestigio/asunto correspondiente.

5.5.2.4. Caso 4: Sincronización masiva offline (REST batch)

Caso de uso: Dispositivo estuvo horas/días sin conexión, acumuló decenas/cientos de eventos, sincroniza al reconectar.

Endpoint: POST /api/v1/sync/upload

Request:

- **Body**: Multipart/form-data con:
  - metadata.json: Array de eventos en formato JSON:

```
[
 {
  "eventId": "evt-123",
  "eventType": "vestigio.creado",
  "timestamp": "2026-01-20T08:15:00Z",
  "data": { /* vestigio JSON */},
  "attachments": ["file1.jpg", "file2.mp4"],
  "checksum": "sha256-abc123..."
 },
 {
  "eventId": "evt-124",
  "eventType": "coc.evento",
  "timestamp": "2026-01-20T08:20:00Z",
  "data": { /* evento CoC JSON */},
  "attachments": [],
  "checksum": "sha256-def456..."
 },
 ...
```

```
    ]
```

    - file1.jpg, file2.mp4, ... : Archivos adjuntos referenciados.
- **Header**: Content-Encoding: gzip (todo el payload comprimido).

Response:


```
{
  "status": "completed",
  "summary": {
    "totalEvents": 120,
    "processed": 118,
    "failed": 2,
    "conflicts": 0
  },
  "details": {
    "failed": [
      {
        "eventId": "evt-150",
        "reason": "Vestigio duplicado (checksum coincide con ID existente)",
        "recoverable": false
      },
      {
        "eventId": "evt-175",
        "reason": "Archivo file3.jpg corrupto (checksum no coincide)",
        "recoverable": true,
        "retry": true
      }
    ]
  }
}
```


Procesamiento en THOT:

1. Descomprimir payload.
2. Validar checksums de metadata y archivos.

3. Ordenar eventos por timestamp (respetar orden temporal original de campo).
4. Para cada evento:
    - Verificar si ya existe (por eventId o checksum de contenido) → si existe, skip (idempotencia).
    - Validar esquema JSON.
    - Almacenar en BBDD/MinIO.
    - Publicar en Kafka (con flag offline_sync=true).
    - Registrar en registro inmutable si corresponde.
5. Detectar conflictos (ej. vestigio modificado en campo y en central durante desconexión), aplicar política de resolución.
6. Retornar resumen con éxitos/fallos.

Manejo de fallos:

- Si sincronización falla parcialmente (ej. 2 de 120 eventos fallaron), dispositivo NO reintenta todos, solo los fallidos (usa campo recoverable de respuesta).
- Si fallo es irrecoverable (ej. evento duplicado), se registra en log local pero no se reintenta, se notifica al usuario.

5.5.3. Sincronización de datos maestros (central → campo)

Además de sincronización de eventos de campo a central, hay flujo inverso: **datos maestros de THOT (central) hacia dispositivos de campo** (ej. catálogos, plantillas, configuración).

Casos:

- **Catálogo de tipos de vestigios**: Lista de tipos válidos con descripciones/iconos, actualizada periódicamente en central, se sincroniza a dispositivos.
- **Plantillas de inspección**: Flujos de trabajo predefinidos (ej. "inspección de vehículo", "inspección de vivienda"), creados por expertos en central, descargados por dispositivos.
- **Configuración de dispositivo**: Políticas (ej. calidad de imagen mínima, frecuencia de telemetría GPS), actualizadas remotamente por administradores.

Mecanismo:

- **Pull periódico**: App de campo cada X horas (ej. cada 6 horas) consulta endpoint GET /api/v1/sync/maestros?last_sync_timestamp=....
- THOT retorna datos maestros modificados desde last_sync_timestamp (delta, no todo).
- App actualiza su BBDD local (SQLite) con nuevos datos.

Push urgente (opcional):

- Si cambio en dato maestro es crítico (ej. nueva política de seguridad), THOT puede enviar notificación push vía MQTT a todos los dispositivos: "Actualización urgente disponible, sincronizar ahora".
- App al recibir notificación, inicia sincronización inmediatamente.

5.5.4. SDK de integración Lote 2 – THOT

Para facilitar desarrollo de apps Lote 2, THOT proporciona **SDK** (Software Development Kit) que abstrae complejidad de integración:

Componentes del SDK:

1. **Cliente API REST**:
   - Librerías pre-construidas (Java/Kotlin para Android, Swift para iOS, Python para equipos Linux).
   - Manejo automático de autenticación (mTLS, renovación de tokens).
   - Reintentos automáticos con backoff exponencial.
   - Serialización/deserialización de JSONs según esquemas de F1.3.1.
2. **Cliente MQTT**:
   - Wrapper sobre librerías MQTT estándar (Eclipse Paho).
   - Gestión automática de reconexión.
   - Helper functions para publicar telemetría (sdk.publishGPS(lat, lon)), suscribirse a notificaciones.
3. **Cliente WebRTC** (OpenVidu Client SDK):
   - Integración simplificada de videoconferencia.
   - API de alto nivel: sdk.startRemoteAssistance(caseId).
4. **Módulo de sincronización offline**:
   - Queue manager para eventos offline (encolar, persistir, sincronizar).
   - Política de retry configurable.
   - Detección automática de conectividad (listener de red).
5. **Utilidades de validación**:
   - Funciones para validar JSONs contra esquemas de F1.3.1 localmente (antes de enviar a THOT).
   - Cálculo de checksums (SHA-256).

Ejemplo de uso (pseudocódigo Kotlin para Android):

```
// Inicializar SDK
val thot = ThotSDK.init(
    context = applicationContext,
    serverUrl = "https://thot.policia.es",
    clientCertificate = loadClientCert()
)

// Registrar vestigio
```


Cofinanciado por la Unión Europea
Fondos Europeos

```
val vestigio = Vestigio(
    asuntoId = "ASU-2026-00123",
    tipoVestigio = TipoVestigio.HUELLA_DACTILAR,
    fechaHoraCaptura = Instant.now(),
    ubicacionCaptura = getGPSLocation(),
    archivo = capturedImageFile
)

thot.vestigios.create(vestigio) { result ->
    when (result) {
        is Success -> {
            showToast("Vestigio registrado: ${result.data.vestigioId}")
            // Evento se registró en central O se encoló si offline
        }
        is Error -> {
            showToast("Error: ${result.message}")
        }
    }
}

// Publicar telemetría GPS periódicamente
thot.telemetry.startGPSUpdates(intervalSeconds = 30)

// Escuchar notificaciones de central
thot.notifications.subscribe { notification ->
    when (notification.type) {
        NotificationType.REMOTE_ASSISTANCE_REQUEST -> {
            showRemoteAssistanceDialog(notification)
        }
        NotificationType.NEW_CASE_ASSIGNED -> {
            refreshCaseList()
        }
    }
```

```
}
```

Distribución del SDK:

- **Repositorio Maven/Gradle** (Android/Java): Artefactos publicados en repositorio interno de THOT, apps Lote 2 añaden dependencia en build.gradle.
- **CocoaPods** (iOS): Pod publicado en repositorio interno.
- **PyPI** (Python): Paquete thot-sdk instalable con pip install thot-sdk --index-url https://repo.thot.policia.es/pypi/simple.

Versioning del SDK: Sigue SemVer (ej. v1.2.3), compatible con versión de API de THOT correspondiente. Si API THOT cambia (nueva versión v2), se publica SDK v2.x compatible.

5.5.5. Requisitos relacionados de la vista Lote 1-Lote 2

Aspecto de interoperabilidad	Requisitos implementados
APIs REST para CRUD	RF-INTEROP-01, INTEROP-3, F1.3.1 sección 4
MQTT para telemetría	INTEROP-3, F1.3.1 sección 6, HW-L1-5
WebRTC para asistencia remota	RF-INTEROP-05, INS-EJE-5, F1.3.1 sección 8, INT-16
Sincronización offline	RF-INTEROP-01, RNF-INTEROP-04, INTEROP-7
SDK de integración	HW-L1-4 (extensibilidad), INTEROP-3

Tabla 19. . Capacidades clave de interoperabilidad de THOT y su trazabilidad con requisitos del pliego (APIs REST, MQTT, WebRTC, sincronización offline y SDK de integración).

6. MODELO DE INTEGRACIÓN Y APIS

Esta sección detalla los patrones de integración utilizados en THOT, las APIs expuestas con sus especificaciones técnicas, y la estrategia de versionado y documentación.

6.1. Patrones de integración

(sincrónica, asíncrona, mensajería, colas, streaming, gateway como patrón de entrada)
THOT implementa múltiples patrones de integración adaptados a las características de cada tipo de interacción: síncrona/asíncrona, crítica/no crítica, volumen alto/bajo.

6.1.1. Patrón 1: Request-Response síncrono (REST)

Cuándo se usa: Operaciones que requieren respuesta inmediata (ej. consultar vestigio, registrar evidencia con confirmación inmediata).

Características:

- **Protocolo**: HTTP/REST
- **Timeout**: 30 segundos máximo
- **Idempotencia**: Operaciones GET/PUT/DELETE son idempotentes, POST puede no serlo (si crea recurso, múltiples POSTs crean múltiples recursos a menos que se implemente detección de duplicados por checksum/ID externo).
- **Manejo de errores**: Códigos HTTP estándar (200 éxito, 400 error cliente, 500 error servidor), body con detalles en formato RFC 7807.

Ventajas: Simplicidad, garantía de respuesta inmediata, fácil debugging.

Desventajas: No escala bien para operaciones largas (>30s), bloquea cliente mientras espera respuesta.

6.1.2. *Patrón 2: Fire-and-Forget asíncrono (Kafka)*

Cuándo se usa: Notificaciones de eventos que no requieren respuesta inmediata (ej. "vestigio creado", "análisis completado"), procesamiento asíncrono desacoplado.

Características:

- **Protocolo**: Apache Kafka (pub/sub)
- **Garantía de entrega**: At-least-once (QoS 1 equivalente), consumidor debe ser idempotente.
- **Orden**: Garantizado dentro de misma partición de Kafka, no garantizado entre particiones (particionar por `asuntoId` si orden es crítico).
- **Latencia**: Sub-segundo típicamente (ms), puede ser mayor si consumidor está sobrecargado.

Ventajas: Desacoplamiento total (productor no sabe quién consume), alta escalabilidad, resiliente (eventos no se pierden aunque consumidor caiga).

Desventajas: No hay confirmación inmediata de procesamiento, debugging más complejo (trazar evento a través de múltiples consumidores).

6.1.3. *Patrón 3: Request-Callback asíncrono (REST + Webhook)*

Cuándo se usa: Operaciones largas que no pueden completarse en <30s (ej. análisis forense complejo que tarda minutos/horas).

Flujo:

1. Cliente hace POST a endpoint de THOT (ej. `POST /api/v1/analisis/adn` para solicitar análisis de ADN).
2. THOT retorna inmediatamente HTTP 202 Accepted + `{jobId}` (ID de trabajo encolado).
3. Cliente puede:

- **Polling**: Consultar periódicamente estado con GET /api/v1/analisis/jobs/{jobId} (retorna {status: "pending" | "running" | "completed" | "failed", progress: 75}).
- **Webhook**: Si cliente proporcionó callbackUrl en petición original, THOT hace POST a esa URL cuando análisis completa (payload: resultado completo).

4. Al completar, resultado queda disponible en GET /api/v1/analisis/jobs/{jobId}/resultado.

Ventajas: Cliente no queda bloqueado esperando, puede hacer otras cosas, se notifica cuando resultado está listo.

Desventajas: Mayor complejidad en cliente (debe manejar async), si usa webhook, cliente debe exponer endpoint HTTP accesible por THOT.

6.1.4. Patrón 4: Streaming bidireccional (gRPC / WebRTC)

Cuándo se usa:

- **gRPC**: Comunicación eficiente de bajo nivel entre microservicios internos (no expuesto a exterior típicamente).
- **WebRTC**: Streaming de vídeo/audio en tiempo real (asistencia remota).

Características gRPC:

- **Protocolo**: HTTP/2 + Protocol Buffers (serialización binaria eficiente)
- **Modos**: Unary (request-response), Server streaming (servidor envía múltiples respuestas), Client streaming (cliente envía múltiples peticiones), Bidirectional streaming (ambos sentidos simultáneos).
- **Ventajas**: Menor latencia que REST, menor uso de ancho de banda (protobuf más compacto que JSON), soporte nativo de streaming.

Uso en THOT: Considerado para comunicación entre microservicio de IA y adaptadores externos si se requiere streaming de resultados parciales (ej. reconocimiento facial progresivo conforme procesa frames de vídeo).

6.1.5. Patrón 5: Adaptador (Adapter Pattern)

Cuándo se usa: Integración con sistemas externos heterogéneos que no siguen protocolos estándar de THOT.

Implementación: Microservicio adaptador específico por sistema externo, traduce entre protocolo de THOT (REST/JSON canónico) y protocolo del sistema externo (SOAP/XML, protocolo propietario, etc.).

Ejemplo: AdaptadorABIS traduce peticiones REST de THOT a SOAP/XML para sistema ABIS legacy, retorna respuesta XML traducida a JSON canónico THOT.

Cofinanciado por la Unión Europea
Fondos Europeos

Ventajas: Aísla complejidad de integración en un componente, resto de THOT no necesita conocer detalles del sistema externo.

6.2. Definición de interfaces y APIs

(RESTful, contratos, endpoints, verbos, formatos)

6.2.1. API de Gestión de Evidencias (REST)

Base URL: https://thot.policia.es/api/v1

Autenticación: mTLS (dispositivos Lote 2, sistemas externos) o OAuth 2.0 Bearer token (usuarios humanos)

6.2.1.1. Endpoints principales

6.2.1.1.1. POST /vestigios - Registrar vestigio

Request:

```
POST /api/v1/vestigios HTTP/1.1
Host: thot.policia.es
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary
Authorization: Bearer <JWT> | (mTLS client cert)

------WebKitFormBoundary
Content-Disposition: form-data; name="metadata"
Content-Type: application/json

{
  "asuntoId": "ASU-2026-00123",
  "tipoVestigio": "huella_dactilar",
  "descripcion": "Huella dactilar latente en vaso",
  "fechaHoraCaptura": "2026-01-21T14:30:00Z",
  "ubicacionCaptura": {
    "direccion": "Calle Gran Vía 123, Madrid",
    "coordenadas": {
```

```
      "latitud": 40.4168,
      "longitud": -3.7038
    }
  },
  "capturadoPor": "Juan Pérez (TIP-123)",
  "checksumArchivo": "a1b2c3d4..."
}
------WebKitFormBoundary
Content-Disposition: form-data; name="archivo"; filename="huella001.jpg"
Content-Type: image/jpeg

<binary data>
------WebKitFormBoundary--
```

Response (éxito):

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/v1/vestigios/VES-2026-00001234

{
  "vestigioId": "VES-2026-00001234",
  "asuntoId": "ASU-2026-00123",
  "fechaHoraRegistro": "2026-01-21T14:30:05Z",
  "url": "https://thot.policia.es/api/v1/vestigios/VES-2026-00001234",
  "coc": {
    "eventoId": "COC-EVT-999888",
    "transaccionRegistroInmutable": "0xabc123..."
  }
}
```

Response (error validación):

```
HTTP/1.1 400 Bad Request
Content-Type: application/problem+json
```

```
{
  "type": "https://thot.policia.es/errors/validation-error",
  "title": "Validation Error",
  "status": 400,
  "detail": "El campo 'asuntoId' es obligatorio y no está presente.",
  "instance": "/api/v1/vestigios",
  "errors": [
    {
      "field": "asuntoId",
      "message": "Campo obligatorio faltante"
    }
  ]
}
```

Criterios de aceptación:

- Validar JSON contra esquema de F1.3.1.
- Verificar checksum de archivo recibido coincide con checksumArchivo declarado.
- Asignar ID único secuencial (formato VES-YYYY-NNNNNNNN).
- Almacenar metadata en PostgreSQL, archivo en MinIO.
- Publicar evento vestigio.creado en Kafka.
- Registrar evento de custodia en registro inmutable
- Latencia p95 <2 segundos (archivo <10 MB, red 4G normal).

6.2.1.1.2. GET /vestigios/{id} - Consultar vestigio

Request:

```
GET /api/v1/vestigios/VES-2026-00001234 HTTP/1.1
Host: thot.policia.es
Authorization: Bearer <JWT>
```

Response (éxito):

```
HTTP/1.1 200 OK
Content-Type: application/json
```

ETag: "a1b2c3d4"

```
{
  "vestigioId": "VES-2026-00001234",
  "asuntoId": "ASU-2026-00123",
  "tipoVestigio": "huella_dactilar",
  "descripcion": "Huella dactilar latente en vaso",
  "fechaHoraCaptura": "2026-01-21T14:30:00Z",
  "fechaHoraRegistro": "2026-01-21T14:30:05Z",
  "ubicacionCaptura": { /* ... */},
  "capturadoPor": "Juan Pérez (TIP-123)",
  "archivos": [
    {
      "archivoId": "FILE-001",
      "nombre": "huella001.jpg",
      "mimeType": "image/jpeg",
      "tamano": 1048576,
      "checksum": "sha256:a1b2c3d4...",
      "urlDescarga": "https://thot.policia.es/api/v1/archivos/FILE-001"
    }
  ],
  "analisis": [
    {
      "analisisId": "ANA-001",
      "tipo": "cotejo_biometrico",
      "estado": "completado",
      "fechaHora": "2026-01-21T14:35:00Z",
      "urlResultado": "/api/v1/analisis/ANA-001"
    }
  ],
  "_links": {
    "self": {"href": "/api/v1/vestigios/VES-2026-00001234"},
    "asunto": {"href": "/api/v1/asuntos/ASU-2026-00123"},
    "coc": {"href": "/api/v1/coc/vestigio/VES-2026-00001234/historial"}
```

```
  }
}
```

Response (no encontrado):

```
HTTP/1.1 404 Not Found
Content-Type: application/problem+json

{
  "type": "https://thot.policia.es/errors/resource-not-found",
  "title": "Resource Not Found",
  "status": 404,
  "detail": "No se encontró vestigio con ID 'VES-2026-00001234'."
}
```

Criterios de aceptación:

- Verificar permisos: usuario puede acceder solo a vestigios de casos asignados (RBAC + row-level security).
- Soportar caché HTTP (headers ETag, If-None-Match, retornar 304 Not Modified si no cambió).
- Latencia p95 <500 ms.

6.2.1.1.3. PATCH /vestigios/{id} - Actualizar parcialmente vestigio

Permite modificar campos específicos sin enviar objeto completo (más eficiente para actualizaciones pequeñas).

Request:

```
PATCH /api/v1/vestigios/VES-2026-00001234 HTTP/1.1
Host: thot.policia.es
Content-Type: application/json
Authorization: Bearer <JWT>
If-Match: "a1b2c3d4"

{
  "descripcion": "Huella dactilar latente en vaso (zona superior)",
  "observaciones": "Huella parcial, calidad media"
```

```
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
ETag: "e5f6g7h8"

{
  "vestigioId": "VES-2026-00001234",
  "descripcion": "Huella dactilar latente en vaso (zona superior)",
  "observaciones": "Huella parcial, calidad media",
  "fechaHoraUltimaModificacion": "2026-01-21T15:00:00Z",
  "modificadoPor": "María López (ANA-456)"
}
```

Criterios de aceptación:

- Validar header If-Match (ETag) para prevenir conflictos de escritura concurrente (optimistic locking).
- Registrar modificación en cadena de custodia (quién, qué campos, cuándo).
- Solo campos permitidos modificables (ej. descripcion, observaciones), campos críticos (checksums, fechas de captura) inmutables.
- Publicar evento vestigio.actualizado en Kafka.

6.2.2. API de Identificación Biométrica (REST)

Base URL: https://thot.policia.es/api/v1/biometria

6.2.2.1.1. POST /biometria/cotejos/huella-dactilar - Solicitar cotejo de huella

Request:

```
POST /api/v1/biometria/cotejos/huella-dactilar HTTP/1.1
Host: thot.policia.es
Content-Type: application/json
Authorization: Bearer <JWT>
```

```
{
  "vestigioId": "VES-2026-00001234",
  "imagenHuella": "<base64-encoded-WSQ>",
  "modoComparacion": "1:N",
  "umbralScore": 8000,
  "callbackUrl": "https://thot.policia.es/api/v1/callbacks/cotejos"
}
```

Response (trabajo encolado):

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: /api/v1/biometria/cotejos/COT-2026-00500

{
  "cotejoId": "COT-2026-00500",
  "estado": "pending",
  "estimacionTiempoSegundos": 10,
  "urlEstado": "/api/v1/biometria/cotejos/COT-2026-00500"
}
```

Consulta estado (polling):

```
GET /api/v1/biometria/cotejos/COT-2026-00500 HTTP/1.1
```

Response (completado):

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "cotejoId": "COT-2026-00500",
  "estado": "completed",
  "fechaHoraInicio": "2026-01-21T15:00:00Z",
```

```
  "fechaHoraFin": "2026-01-21T15:00:08Z",
  "candidatos": [
    {
      "personaId": "PER-12345678",
      "nif": "12345678A",
      "nombre": "Juan García",
      "score": 9850,
      "imagenReferencia": "https://thot.policia.es/api/v1/personas/PER-12345678/huella",
      "metadata": {
        "dedoReferencia": "indice_derecho",
        "fechaRegistroReferencia": "2020-05-10"
      }
    },
    {
      "personaId": "PER-87654321",
      "nif": "87654321B",
      "nombre": "Pedro Martínez",
      "score": 8500,
      "imagenReferencia": "https://thot.policia.es/api/v1/personas/PER-87654321/huella",
      "metadata": { /* ... */ }
    }
  ],
  "metadata": {
    "totalCandidatos": 2,
    "tiempoProcesamientoMs": 8234,
    "sistemaABIS": "ABIS-PROD-v3.2"
  }
}
```


Criterios de aceptación:

- Validar formato de imagen (WSQ preferido, JPEG/PNG aceptados pero convertidos a WSQ).
- Encollar petición en sistema de colas (Redis Queue o similar).
- Adaptador ABIS procesa de forma asíncrona.
- Si callbackUrl proporcionado, hacer POST a esa URL al completar (con retry si falla).
- Almacenar resultado en caché (Redis, TTL 1 hora).

- Registrar consulta biométrica en log de auditoría (cumplimiento RGPD: quién consultó qué datos biométricos, cuándo).
- Latencia total p95 <15 segundos (incluye tiempo de cotejo en ABIS).

6.2.3. API de Canal de Eventos (WebSocket / Server-Sent Events)

Para notificaciones push en tiempo real desde THOT a clientes web (alternativa a polling).

Protocolo: WebSocket (bidireccional) o Server-Sent Events (SSE, unidireccional servidor → cliente).

Endpoint: wss://thot.policia.es/api/v1/eventos/stream

Flujo:

1. Cliente establece conexión WebSocket (autenticación vía header Authorization o query param ?token=<JWT>).
2. Cliente se suscribe a canales de interés enviando mensaje:

```
{
  "action": "subscribe",
  "channels": ["vestigios.actualizados", "analisis.completados", "alertas"]
}
```

3. Servidor confirma suscripción.
4. Cuando ocurre evento relevante (ej. análisis completa), servidor envía mensaje a todos los clientes suscritos a ese canal:

```
{
  "channel": "analisis.completados",
  "event": {
    "analisisId": "ANA-001",
    "vestigioId": "VES-2026-00001234",
    "tipo": "cotejo_biometrico",
    "estado": "completado",
    "timestamp": "2026-01-21T15:00:08Z"
  }
}
```

5. Cliente recibe mensaje, actualiza UI sin necesidad de polling.

Ventajas sobre polling: Menor latencia (push inmediato vs polling cada N segundos), menor carga en servidor (no cientos de clientes haciendo GET cada segundo).

Consideraciones de escalabilidad: Si miles de clientes conectados simultáneamente, usar Redis Pub/Sub como backend para distribuir eventos entre múltiples instancias del servicio WebSocket.

6.3. Versionado de APIs y documentación

(OpenAPI/Swagger, catálogos de servicios)
THOT aplica **versionado semántico (SemVer)** adaptado a APIs REST:

Formato de versión: v{MAJOR}.{MINOR}.{PATCH} (ej. v1.0.0, v1.2.3, v2.0.0)

Incremento de versiones:

- **MAJOR** (ej. v1 → v2): Cambios incompatibles hacia atrás (breaking changes). Ej: eliminar endpoint, cambiar tipo de dato de campo, cambiar semántica de operación.
- **MINOR** (ej. v1.0 → v1.1): Añadir funcionalidad compatible. Ej: nuevo endpoint, nuevos campos opcionales en request/response, nuevos valores en enum.
- **PATCH** (ej. v1.0.0 → v1.0.1): Corrección de bugs sin cambio de funcionalidad. Ej: corregir error 500, mejorar mensaje de error.

Estrategia de versionado en URL: Major version en path (/api/v1/, /api/v2/), minor/patch no aparecen en URL (se documentan en header response API-Version: 1.2.3).

Ejemplo:

- Cliente hace GET /api/v1/vestigios/123.
- Servidor retorna header API-Version: 1.2.3, indicando que versión exacta de implementación es 1.2.3, pero cliente puede usar cualquier cliente compatible con v1.

Política de deprecación:

1. Cuando se introduce v2, v1 se marca como **deprecated** (no removida inmediatamente).
2. Se documenta en:
    - Header HTTP Warning: 299 - "API v1 is deprecated. Migrate to v2 by 2027-01-21. See https://thot.policia.es/docs/api/migration-v1-to-v2".
    - OpenAPI spec con campo deprecated: true en endpoints afectados.
    - Notificación por email a clientes registrados (desarrolladores Lote 2, sistemas externos).
3. **Periodo de transición**: Mínimo 6 meses donde v1 y v2 coexisten.
4. Tras 6 meses, v1 se elimina: peticiones a /api/v1/* retornan HTTP 410 Gone con mensaje indicando migrar a v2.

Excepciones: Si cambio es crítico de seguridad y no puede esperar 6 meses, periodo de transición se reduce a 1 mes con notificación urgente.

Cofinanciado por la Unión Europea
Fondos Europeos

6.4. Documentación y portal de desarrolladores

6.4.1. Portal de Desarrolladores THOT

URL: https://developers.thot.policia.es

Contenido:

1. **Getting Started**: Guía de inicio rápido para integrar con THOT.
    - Cómo obtener credenciales (certificado mTLS o API Key).
    - Ejemplo "Hello World" (registrar vestigio con cURL).
    - SDK disponibles (Android, iOS, Python).
2. **API Reference**: Documentación completa de APIs generada automáticamente desde especificaciones OpenAPI.
    - Cada endpoint con descripción, parámetros, request/response examples, códigos de error.
    - Interfaz interactiva (Swagger UI / Redoc) permite probar endpoints desde navegador (sandbox environment).
3. **Guides**: Tutoriales paso a paso para casos de uso comunes.
    - "Cómo registrar y consultar un vestigio"
    - "Cómo solicitar un cotejo biométrico"
    - "Cómo implementar sincronización offline"
    - "Cómo integrar asistencia remota WebRTC"
4. **SDKs and Tools**: Descarga de SDKs, librerías auxiliares, herramientas de testing.
    - SDK Android (Kotlin/Java)
    - SDK iOS (Swift)
    - SDK Python
    - CLI tool para operaciones desde terminal
    - Postman collection con ejemplos de peticiones
5. **Changelog**: Historial de cambios en APIs.
    - Qué cambió en cada versión (features añadidas, bugs corregidos, breaking changes).
    - Migration guides entre versiones major.
6. **Status Page**: Estado en tiempo real de APIs de THOT.
    - Indicadores de disponibilidad (uptime), latencia (p50/p95), tasa de error.
    - Notificaciones de incidencias en curso o mantenimientos programados.
7. **Support**: Canales de soporte técnico.
    - FAQ (preguntas frecuentes).
    - Foro comunitario (Q&A entre desarrolladores).
    - Email de soporte: api-support@thot.policia.es.
    - Issue tracker (para reportar bugs en APIs).

Cofinanciado por la Unión Europea
Fondos Europeos

6.4.2. Especificaciones OpenAPI

Cada API REST de THOT tiene especificación **OpenAPI 3.0** completa, publicada en:

- **Formato YAML/JSON**: https://thot.policia.es/api/v1/openapi.json
- **Documentación HTML interactiva**: https://thot.policia.es/api/v1/docs (Swagger UI)

Ejemplo de fragmento OpenAPI (vestigios endpoint):

```yaml
openapi: 3.0.3
info:
  title: THOT Interoperability API
  version: 1.2.3
  description: API de interoperabilidad de THOT para integración con Lote 2 y sistemas externos
  contact:
    name: Equipo API THOT
    email: api-support@thot.policia.es
servers:
  - url: https://thot.policia.es/api/v1
    description: Producción
  - url: https://thot-pre.policia.es/api/v1
    description: Preproducción

paths:
  /vestigios:
    post:
      summary: Registrar nuevo vestigio
      operationId: crearVestigio
      tags:
        - Vestigios
      security:
        - mTLS: []
        - OAuth2: [write:vestigios]
      requestBody:
        required: true
```

```
    content:
      multipart/form-data:
        schema:
          type: object
          required:
            - metadata
            - archivo
          properties:
            metadata:
              $ref: '#/components/schemas/VestigioMetadata'
            archivo:
              type: string
              format: binary
              description: Archivo binario del vestigio (imagen, vídeo, documento)
  responses:
    '201':
      description: Vestigio creado exitosamente
      headers:
        Location:
          schema:
            type: string
            example: /api/v1/vestigios/VES-2026-00001234
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/VestigioCreado'
    '400':
      description: Error de validación
      content:
        application/problem+json:
          schema:
            $ref: '#/components/schemas/ProblemDetails'
    '401':
      description: No autenticado
```

```
        '403':
          description: No autorizado (sin permisos)
        '500':
          description: Error interno del servidor

components:
  schemas:
    VestigioMetadata:
      type: object
      required:
        - asuntoId
        - tipoVestigio
        - fechaHoraCaptura
        - capturadoPor
        - checksumArchivo
      properties:
        asuntoId:
          type: string
          pattern: '^ASU-[0-9]{4}-[0-9]{5,8}$'
          example: ASU-2026-00123
        tipoVestigio:
          type: string
          enum: [huella_dactilar, huella_palmar, adn_biologico, fibra_textil, proyectil_balistica,
documento_papel, dispositivo_electronico, sustancia_quimica, imagen_fotografica, video_grabacion,
audio_grabacion, otro]
        descripcion:
          type: string
          maxLength: 500
        fechaHoraCaptura:
          type: string
          format: date-time
          description: Timestamp ISO 8601 en UTC
        ubicacionCaptura:
          $ref: '#/components/schemas/Ubicacion'
```

```
    capturadoPor:
      type: string
      description: Nombre y identificador del técnico que capturó
    checksumArchivo:
      type: string
      pattern: '^[a-f0-9]{64}$'
      description: Hash SHA-256 del archivo en hexadecimal

VestigioCreado:
  type: object
  properties:
    vestigioId:
      type: string
      example: VES-2026-00001234
    asuntoId:
      type: string
    fechaHoraRegistro:
      type: string
      format: date-time
    url:
      type: string
      format: uri
    coc:
      type: object
      properties:
        eventoId:
          type: string
        transaccionRegistroInmutable:
          type: string

Ubicacion:
  type: object
  properties:
    direccion:
```

```
        type: string
      coordenadas:
        type: object
        required:
          - latitud
          - longitud
        properties:
          latitud:
            type: number
            minimum: -90
            maximum: 90
          longitud:
            type: number
            minimum: -180
            maximum: 180

  ProblemDetails:
    type: object
    description: RFC 7807 Problem Details for HTTP APIs
    required:
      - type
      - title
      - status
    properties:
      type:
        type: string
        format: uri
        example: https://thot.policia.es/errors/validation-error
      title:
        type: string
        example: Validation Error
      status:
        type: integer
        example: 400
```

```
      detail:
        type: string
        example: El campo 'asuntoId' es obligatorio.
      instance:
        type: string
        format: uri
        example: /api/v1/vestigios
      errors:
        type: array
        items:
          type: object
          properties:
            field:
              type: string
            message:
              type: string

securitySchemes:
  mTLS:
    type: mutualTLS
    description: Autenticación mutua TLS con certificado cliente X.509
  OAuth2:
    type: oauth2
    flows:
      authorizationCode:
        authorizationUrl: https://idp.policia.es/oauth2/authorize
        tokenUrl: https://idp.policia.es/oauth2/token
        scopes:
          read:vestigios: Leer vestigios
          write:vestigios: Crear/modificar vestigios
          read:analisis: Leer resultados de análisis
          write:analisis: Solicitar análisis
```

Generación automática: Las especificaciones OpenAPI se generan automáticamente desde anotaciones en código de microservicios (ej. decoradores en Spring Boot, FastAPI) y se publican en cada despliegue, garantizando que documentación está siempre sincronizada con implementación real.

7. INTERCAMBIO DE DATOS Y SEMÁNTICA

Esta sección profundiza en los modelos de datos intercambiados, estrategias de normalización semántica, gestión de archivos binarios y metadatos de trazabilidad.

7.1. Modelos de datos compartidos y esquemas

Todos los datos estructurados intercambiados entre THOT y sistemas externos (Lote 2, TIC internos) se validan contra **JSON Schema Draft-07**.

7.1.1. Esquema: Vestigio

(Definido en F1.3.1, reproducido aquí para referencia)

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://thot.policia.es/schemas/vestigio.json",
  "title": "Vestigio",
  "description": "Evidencia física o digital capturada en una investigación",
  "type": "object",
  "required": ["asuntoId", "tipoVestigio", "fechaHoraCaptura", "capturadoPor"],
  "properties": {
    "vestigioId": {
      "type": "string",
      "pattern": "^VES-[0-9]{4}-[0-9]{8}$",
      "description": "Identificador único asignado por THOT"
    },
    "asuntoId": {
      "type": "string",
      "pattern": "^ASU-[0-9]{4}-[0-9]{5,8}$",
      "description": "ID del asunto al que pertenece"
```

```
    },
    "tipoVestigio": {
      "type": "string",
      "enum": ["huella_dactilar", "huella_palmar", "adn_biologico", "fibra_textil", "proyectil_balistica",
"documento_papel", "dispositivo_electronico", "sustancia_quimica", "imagen_fotografica",
"video_grabacion", "audio_grabacion", "otro"]
    },
    "descripcion": {
      "type": "string",
      "maxLength": 500
    },
    "fechaHoraCaptura": {
      "type": "string",
      "format": "date-time",
      "description": "Timestamp ISO 8601 UTC"
    },
    "ubicacionCaptura": {
      "$ref": "#/definitions/Ubicacion"
    },
    "capturadoPor": {
      "type": "string",
      "description": "Nombre y credencial del técnico"
    },
    "archivos": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Archivo"
      },
      "minItems": 1
    },
    "metadataAdicional": {
      "type": "object",
      "description": "Metadatos específicos del tipo de vestigio",
      "additionalProperties": true
```

```
    }
  },
  "definitions": {
    "Ubicacion": {
      "type": "object",
      "properties": {
        "direccion": {"type": "string"},
        "coordenadas": {
          "type": "object",
          "required": ["latitud", "longitud"],
          "properties": {
            "latitud": {"type": "number", "minimum": -90, "maximum": 90},
            "longitud": {"type": "number", "minimum": -180, "maximum": 180}
          }
        }
      }
    },
    "Archivo": {
      "type": "object",
      "required": ["nombre", "mimeType", "tamano", "checksum"],
      "properties": {
        "archivoId": {"type": "string"},
        "nombre": {"type": "string"},
        "mimeType": {"type": "string"},
        "tamano": {"type": "integer", "minimum": 0},
        "checksum": {"type": "string", "pattern": "^sha256:[a-f0-9]{64}$"},
        "urlDescarga": {"type": "string", "format": "uri"}
      }
    }
  }
}
```

7.1.2. Esquema: Persona (integración con sistema PERSONAS)

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://thot.policia.es/schemas/persona.json",
  "title": "Persona",
  "description": "Datos de filiación de una persona",
  "type": "object",
  "required": ["personaId", "nif", "nombre", "apellidos"],
  "properties": {
    "personaId": {
      "type": "string",
      "pattern": "^PER-[0-9]{8}$"
    },
    "nif": {
      "type": "string",
      "pattern": "^[0-9]{8}[A-Z]$",
      "description": "NIF español sin guiones"
    },
    "nie": {
      "type": "string",
      "pattern": "^[XYZ][0-9]{7}[A-Z]$",
      "description": "NIE (extranjeros residentes)"
    },
    "nombre": {
      "type": "string"
    },
    "apellidos": {
      "type": "string"
    },
    "fechaNacimiento": {
      "type": "string",
      "format": "date"
```

```
    },
    "lugarNacimiento": {
      "type": "string"
    },
    "sexo": {
      "type": "string",
      "enum": ["M", "F", "X"]
    },
    "nacionalidad": {
      "type": "string",
      "description": "Código ISO 3166-1 alpha-3",
      "pattern": "^[A-Z]{3}$"
    },
    "direccion": {
      "type": "string"
    },
    "fotografia": {
      "type": "string",
      "format": "uri",
      "description": "URL de fotografía oficial"
    },
    "datosBiometricos": {
      "type": "object",
      "properties": {
        "huellasDactilares": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "dedo": {"type": "string", "enum": ["pulgar_derecho", "indice_derecho", /*... */]},
              "imagen": {"type": "string", "format": "uri"},
              "minucias": {"type": "string", "description": "Minucias en formato ANSI/NIST"}
            }
          }
```


Cofinanciado por la Unión Europea
Fondos Europeos

```
      },
      "iris": {"type": "string", "format": "uri"},
      "adn": {
        "type": "object",
        "properties": {
          "perfilGenetico": {"type": "string", "description": "Perfil STR"},
          "laboratorio": {"type": "string"}
        }
      }
    }
  }
}
```

7.1.3. Esquema: Evento de Cadena de Custodia

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://thot.policia.es/schemas/evento-coc.json",
  "title": "EventoCadenaCustodia",
  "description": "Evento en la cadena de custodia de una evidencia",
  "type": "object",
  "required": ["vestigioId", "tipoEvento", "fechaHora", "responsable"],
  "properties": {
    "eventoId": {
      "type": "string",
      "pattern": "^COC-EVT-[0-9]{6,}$"
    },
    "vestigioId": {
      "type": "string"
    },
    "tipoEvento": {
      "type": "string",
```


Cofinanciado por la Unión Europea
Fondos Europeos

```
      "enum": ["captura", "transferencia", "acceso", "analisis", "almacenamiento", "destruccion"]
    },
    "fechaHora": {
      "type": "string",
      "format": "date-time"
    },
    "responsable": {
      "type": "string",
      "description": "Nombre y credencial del responsable"
    },
    "ubicacion": {
      "type": "string",
      "description": "Lugar físico donde ocurrió el evento"
    },
    "descripcion": {
      "type": "string"
    },
    "metadataEvento": {
      "type": "object",
      "description": "Datos específicos del tipo de evento",
      "additionalProperties": true
    },
    "transaccionRegistroInmutable": {
      "type": "string",
      "description": "ID de transacción en Hyperledger Fabric"
    }
  }
}
```


7.2. Transformación y mapeo de datos entre sistemas internas

(ETL/ELT, adaptadores, normalización entre bases de datos internas de la policía: CODIS, ABIS y IBIN, etc.)

7.2.1. Mapeo desde sistemas TIC internos

Los sistemas TIC internos de la Policía Nacional (PERSONAS, ABIS, etc.) pueden tener modelos de datos propios que no coinciden exactamente con esquemas canónicos de THOT. Los adaptadores realizan **mapeo semántico**:

Ejemplo: Mapeo de datos de persona desde sistema PERSONAS a THOT

Campo en sistema PERSONAS	Campo canónico THOT	Transformación
DNI_NUMERO	nif	Normalizar: quitar guiones, convertir letra a mayúscula
NOMBRE_COMPLETO	nombre, apellidos	Dividir string por espacios, asumir primer token=nombre, resto=apellidos (puede requerir lógica más sofisticada)
FECHA_NAC (formato DD/MM/YYYY)	fechaNacimiento (formato YYYY-MM-DD)	Parsear y reformatear
SEXO (H/M/I)	sexo (M/F/X)	Mapear: H→M, M→F, I→X
PAIS_NACIONALIDAD (nombre completo ej. "España")	nacionalidad (código ISO, ej. "ESP")	Lookup en tabla de países ISO 3166

Tabla 20. Ejemplo de mapeo y transformación de campos entre un sistema externo (PERSONAS) y el modelo canónico de THOT, incluyendo normalización, parsing de formatos, reglas de mapeo y enriquecimiento mediante códigos ISO.

Implementación: Adaptador mantiene configuración de mapeo en archivo YAML/JSON, permite ajustar mapeo sin cambiar código:

```yaml
# config/mapeo-personas.yaml
mappings:
  - source: DNI_NUMERO
    target: nif
    transform: normalize_nif
  - source: NOMBRE_COMPLETO
    target: [nombre, apellidos]
    transform: split_name
  - source: FECHA_NAC
```

```
  target: fechaNacimiento
  transform: parse_date
  sourceFormat: DD/MM/YYYY
  targetFormat: YYYY-MM-DD
```

7.2.2. Enriquecimiento semántico con JSON-LD

Para facilitar interoperabilidad con sistemas que usan **Linked Data** (ej. bases de datos policiales europeas que adopten W3C standards), THOT puede exportar datos en formato JSON-LD.

Contexto JSON-LD para vestigios:

```
{
  "@context": {
    "@vocab": "https://thot.policia.es/ontology/forense#",
    "dc": "http://purl.org/dc/terms/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "geo": "http://www.w3.org/2003/01/geo/wgs84_pos#",
    "prov": "http://www.w3.org/ns/prov#",

    "Vestigio": "@type",
    "vestigioId": "@id",
    "asuntoId": {"@type": "@id"},
    "tipoVestigio": {"@type": "xsd:string"},
    "descripcion": {"@type": "xsd:string"},
    "fechaHoraCaptura": {"@type": "xsd:dateTime"},
    "capturadoPor": {"@id": "prov:wasAttributedTo"},
    "ubicacionCaptura": {
      "@id": "geo:location",
      "@type": "geo:Point"
    }
  }
}
```

Ventajas:

- Sistemas externos pueden interpretar datos sin necesitar esquema específico de THOT (leen contexto).
- Permite queries semánticas (SPARQL) si datos se cargan en triplestore RDF.

Obligatoriedad: JSON-LD es **opcional** para integraciones internas THOT-Lote 2 (suficiente con JSON Schema). **Recomendado** para exportación a sistemas internacionales (INTERPOL, EUROPOL, EURODAC) si requieren semántica formal.

7.3. Transformación y mapeo de datos con sistemas externos

(Automatización de procesos de intercambio de información con bases de datos externas: INTERPOL, EUROPOL, SIS RECAST y SMART BORDERS)

7.3.1. Formatos aceptados

Tipo de evidencia	Formatos aceptados	Formato preferido	Conversión automática
Imagen forense	JPEG, PNG, TIFF, RAW (CR2/NEF/ARW), WSQ (huellas)	WSQ (huellas), TIFF sin compresión (general)	PNG/JPEG → TIFF si calidad crítica
Vídeo	MP4 (H.264/H.265), MOV, AVI, MKV	MP4 con H.265 (mejor compresión)	Códecs legacy → H.265 en background
Audio	MP3, WAV, FLAC, AAC, OGG	FLAC (sin pérdida) o AAC (con pérdida aceptable)	MP3 → FLAC si almacenamiento lo permite
Documentos	PDF, DOC/DOCX, XLS/XLSX, TXT, HTML	PDF/A (archival)	DOC/DOCX → PDF/A
Volcado forense	E01 (EnCase), AFF (Advanced Forensic Format), DD (raw image)	E01 (estándar de facto)	-

Tabla 21. Estándares de formato por tipo de evidencia y su normalización automática. *La tabla define, para cada tipo de evidencia forense, los formatos que el sistema acepta, el formato preferente recomendado para almacenamiento/procesado, y si existe conversión automática para homogeneizar y asegurar calidad, compatibilidad y preservación a largo plazo.*

7.3.2. Almacenamiento en Object Storage (MinIO)

Organización por buckets:

- thot-vestigios-prod: Archivos de vestigios (inmutables).
- thot-analisis-prod: Resultados de análisis (informes PDF, imágenes procesadas).
- thot-teleasistencia-prod: Grabaciones de sesiones WebRTC.
- thot-backup: Backups de bases de datos, logs.

Naming convention de objetos:

- Formato: {asuntoId}/{vestigioId}/{timestamp}_{filename}
- Ejemplo: ASU-2026-00123/VES-2026-00001234/20260121T143000Z_huella001.jpg

Metadata de objetos (S3 object tags):

- asuntoId: ASU-2026-00123
- vestigioId: VES-2026-00001234
- tipoVestigio: huella_dactilar
- clasificacionSeguridad: confidencial
- checksumSHA256: a1b2c3d4...
- retentionPolicy: 7years (ISO 21043 recomienda 7 años mínimo)

Cifrado: Server-side encryption (SSE) con AES-256, claves gestionadas por Vault.

Versionado: Habilitado en buckets de vestigios (si archivo se reemplaza accidentalmente, versión anterior recuperable).

WORM (Write Once Read Many): Habilitar en bucket de vestigios para inmutabilidad (cumplimiento ENS Alto + ISO 21043). Una vez escrito, objeto no puede ser modificado ni eliminado durante periodo de retención.

7.3.3. *Thumbnails y previsualizaciones*

Para mejorar rendimiento de UI (evitar descargar archivos completos solo para preview), THOT genera **thumbnails** automáticamente:

Estrategia:

- Al subir imagen/vídeo, microservicio de procesamiento (background job) genera:
  - **Thumbnail pequeño** (150x150 px JPEG, ~10 KB) para listados.
  - **Preview mediano** (800x600 px JPEG, ~100 KB) para vista detalle.
  - Original se mantiene intacto en MinIO.
- Thumbnails se almacenan en bucket separado (thot-thumbnails-prod) con mismo naming convention + sufijo -thumb-150 / -preview-800.
- Metadata de vestigio incluye URLs de thumbnails:

- "archivos": [{
- "archivoId": "FILE-001",
- "urlDescarga": "https://thot.policia.es/api/v1/archivos/FILE-001",
- "urlThumbnail": "https://thot.policia.es/api/v1/archivos/FILE-001/thumbnail",
- "urlPreview": "https://thot.policia.es/api/v1/archivos/FILE-001/preview"
- }]

Generación de thumbnails para vídeos: Extraer frame representativo (ej. frame a 5 segundos del inicio, o frame con mejor calidad de imagen detectado automáticamente).

7.4. Intercambio de datos con la escena

Cada intercambio de datos entre THOT y sistemas externos/Lote 2 genera **metadatos de trazabilidad** que permiten reconstruir historial completo de un dato.

7.4.1. Metadata obligatoria en cada operación


```
{
  "operacionId": "OP-20260121-00123456",
  "timestamp": "2026-01-21T14:30:00.123Z",
  "actor": {
    "tipo": "usuario | dispositivo | sistema",
    "id": "user123 | DEV-00123 | SYS-ABIS",
    "nombre": "Juan Pérez | Tableta-00123 | ABIS-Prod"
  },
  "accion": "create | read | update | delete | query",
  "recurso": {
    "tipo": "vestigio | persona | analisis | coc_evento",
    "id": "VES-2026-00001234"
  },
  "origen": {
    "sistema": "THOT | Lote2 | ABIS | PERSONAS",
    "ip": "192.168.1.100",
    "ubicacion": "Madrid, España"
```

```
  },
  "resultado": "success | failure",
  "detalles": {
    "camposModificados": ["descripcion", "observaciones"],
    "razonFallo": "Permiso denegado" (si resultado=failure)
  },
  "correlacionId": "trace-abc123",
  "sessionId": "sess-xyz789"
}
```

Almacenamiento:

- Metadatos de trazabilidad se almacenan en **ElasticSearch** (indexados, queryables).
- Se exportan diariamente a MinIO como backup inmutable (WORM).
- Eventos críticos (creación/modificación de vestigio, acceso a datos biométricos) se registran también en registro inmutable para inmutabilidad verificable.

Consultas de trazabilidad:

- UI de THOT permite a auditores consultar: "Quién accedió al vestigio VES-2026-00001234 en los últimos 30 días?"
- Query a ElasticSearch retorna lista de operaciones con timestamps, actores, acciones.
- Si necesario demostrar inmutabilidad de log (ej. en proceso judicial), se verifica contra registro inmutable.

8. GESTIÓN OPERATIVA DE LA INTEROPERABILIDAD

Esta sección define cómo se monitoriza, mantiene y evoluciona el sistema de interoperabilidad en producción.

8.1. Monitorización de integraciones

(logs, métricas, alertas)

8.1.1. Métricas de interoperabilidad

Métricas técnicas (recolectadas por Prometheus):

Métrica	Descripción	Umbral	Alerta
api_requests_total	Total de peticiones a APIs (por endpoint)	-	-
api_requests_duration_seconds	Latencia de peticiones (p50, p95, p99)	p95 <2s	Crítica si p95 >5s
api_requests_errors_total	Total de errores (por código HTTP)	<1%	Crítica si >5%
kafka_messages_produced_total	Mensajes publicados en Kafka	-	-
kafka_messages_consumed_total	Mensajes consumidos de Kafka	-	-
kafka_consumer_lag	Lag de consumidores (mensajes pendientes)	<1000	Crítica si >10000
integration_external_success_rate	Tasa de éxito de integraciones externas (ABIS, PERSONAS)	>95%	Crítica si <90%
integration_external_latency_seconds	Latencia de sistemas externos (p95)	Varía por sistema	Alerta si >2x baseline
registro_inmutable_transaction_time_seconds	Tiempo de confirmación de transacción en registro inmutable	p95 <5s	Crítica si >30s
offline_sync_queue_size	Tamaño de cola de sincronización offline	<100	Alerta si >500

Tabla 22. Métricas técnicas de interoperabilidad de THOT recolectadas mediante Prometheus para supervisar el rendimiento y la fiabilidad de las interfaces (APIs REST, mensajería Kafka, integraciones externas, registro inmutable y sincronización offline), incluyendo umbrales objetivo y criterios de alerta para detección temprana de degradaciones y activación de acciones operativas.

Métricas de negocio (calculadas desde datos de BBDD):

Métrica	Descripción	Frecuencia cálculo	Dashboard
vestigios_registrados_por_dia	Número de vestigios	Diaria	Grafana "Operaciones"

Cofinanciado por la Unión Europea
Fondos Europeos

Métrica	Descripción	Frecuencia cálculo	Dashboard
	registrados por día		
analisis_biometricos_completados_por_hora	Throughput de análisis biométricos	Horaria	Grafana "Biometría"
tasa_cotejo_biometrico_exitoso	% de cotejos que encontraron match	Diaria	Grafana "Biometría"
sesiones_teleasistencia_por_dia	Número de sesiones WebRTC	Diaria	Grafana "Teleasistencia"
eventos_coc_registrados	Eventos de cadena de custodia	Tiempo real	Grafana "Cadena Custodia"

Tabla 23. Métricas de negocio de THOT calculadas a partir de datos de base de datos para monitorizar actividad operativa y resultados funcionales (registro de vestigios, rendimiento biométrico, tasa de cotejo, teleasistencia y eventos de Cadena de Custodia), indicando frecuencia de cálculo y su visualización en paneles Grafana por dominio.

8.1.2. Dashboards de monitorización (Grafana)

Dashboard "Salud de Integraciones":

- Panel 1: Tasa de éxito de APIs (% de HTTP 2xx vs 4xx/5xx) en últimas 24h, desglosado por endpoint.
- Panel 2: Latencia p95 de APIs principales (últimas 24h), línea temporal.
- Panel 3: Estado de sistemas externos (ABIS, PERSONAS, etc.): verde/amarillo/rojo según disponibilidad.
- Panel 4: Lag de consumidores Kafka (últimas 24h).
- Panel 5: Alertas activas (lista de alertas disparadas en últimas 24h).

Dashboard "Lote 2 - Campo":

- Panel 1: Dispositivos online (mapa geográfico con marcadores).
- Panel 2: Vestigios registrados desde campo (últimas 24h, desglosado por tipo).
- Panel 3: Cola de sincronización offline (tamaño actual, tendencia).
- Panel 4: Telemetría promedio: nivel de batería de dispositivos, calidad de red (4G/5G), temperatura.

Dashboard "Cadena de Custodia":

- Panel 1: Eventos de custodia registrados (últimas 24h, desglosado por tipo).
- Panel 2: Latencia de registro en registro inmutable (p95).
- Panel 3: Vestigios sin evento de custodia en >24h (alerta de anomalía).

Acceso a dashboards:

- **Operadores**: Acceso read-only a dashboards relevantes (pueden ver, no modificar).
- **Administradores**: Acceso completo (pueden crear/modificar dashboards, configurar alertas).

8.2. Gestión de incidencias y soporte de integraciones

8.2.1. Clasificación de incidencias

Severidad	Definición	Ejemplo	Tiempo respuesta	Tiempo resolución objetivo
Crítica	Sistema completamente inoperativo, imposible registrar evidencias	API Gateway caído, ImmuDB no acepta transacciones, ABIS no responde	<1 hora	<4 horas
Alta	Funcionalidad importante degradada, pero workaround existe	Latencia de ABIS muy alta (>30s), sincronización offline con alta tasa de error	<4 horas	<24 horas
Media	Funcionalidad secundaria afectada	Thumbnails no se generan, logs no se indexan en ElasticSearch	<24 horas	<72 horas
Baja	Problema cosmético o mejora deseada	Mensaje de error poco claro, documentación desactualizada	<72 horas	<1 semana

Tabla 24. Clasificación de incidencias operativas en THOT por nivel de severidad, definiendo el impacto sobre la continuidad del servicio y la operativa de registro de evidencias, con ejemplos representativos y los objetivos de tiempos de respuesta y resolución (SLA/OLA) asociados para priorización y escalado.

8.2.2. Procedimiento de respuesta a incidencias

1. **Detección**:
    - Automática (alerta de Prometheus/Grafana, monitorización sintética).
    - Manual (reporte de usuario vía email/teléfono, ticket en sistema de incidencias INT-17).
2. **Triage** (equipo de guardia 24/7 en producción):
    - Clasificar severidad.
    - Asignar a equipo responsable (backend, integraciones, infraestructura).
    - Crear ticket en sistema de gestión (ej. Jira, ServiceNow) con timeline.
3. **Investigación**:

   - Consultar logs (Kibana), métricas (Grafana), trazas distribuidas (Jaeger).
   - Reproducir problema en entorno de desarrollo/preproducción si es posible.
   - Identificar causa raíz.
4. **Mitigación temporal** (si severidad crítica/alta):
   - Aplicar workaround (ej. si ABIS caído, deshabilitar cotejos automáticos temporalmente, encolar peticiones para procesar cuando se recupere).
   - Comunicar a usuarios afectados (via UI, email, notificación MQTT).
5. **Resolución definitiva**:
   - Implementar fix (código, configuración, infraestructura).
   - Probar en preproducción.
   - Desplegar a producción (siguiendo proceso de cambios, sección 8.3).
   - Verificar que problema está resuelto (monitorizar métricas post-despliegue).
6. **Post-mortem** (para incidencias críticas/altas):
   - Reunión de equipo (1-2 días tras resolución) para analizar:
     - ¿Qué falló?
     - ¿Por qué no lo detectamos antes?
     - ¿Qué acciones correctivas evitarán recurrencia?
   - Documentar en wiki de equipo, compartir aprendizajes.
   - Implementar acciones correctivas (ej. añadir nueva alerta, mejorar validación de datos).

8.2.3. Comunicación durante incidencias

Status page pública (https://status.thot.policia.es):

- Indicadores de estado de servicios (verde/amarillo/rojo).
- Histórico de incidencias (últimos 90 días).
- Suscripción a notificaciones (email/SMS cuando cambia estado).

Notificaciones proactivas a integradores (desarrolladores Lote 2, administradores sistemas externos):

- Email a lista api-status@thot.policia.es cuando incidencia crítica/alta afecta APIs.
- Incluir: descripción del problema, impacto, workaround si existe, ETA de resolución.

8.3. Políticas de cambio y despliegue

(DevOps, CI/CD, entornos)

8.3.1. Política de cambios en APIs

Cambios no disruptivos (no requieren nueva versión major):

- Añadir nuevo endpoint.
- Añadir campo opcional a request/response.

- Añadir nuevo valor a enum (si clientes manejan valores desconocidos correctamente).
- Corregir bug que hacía que API no cumpliera especificación.

Proceso:

- Despliegue directo a producción tras pruebas en preproducción.
- Documentar en Changelog.
- No requiere coordinación con clientes (backwards compatible).

Cambios disruptivos (requieren nueva versión major):

- Eliminar endpoint o campo.
- Cambiar tipo de dato de campo (ej. string → integer).
- Cambiar semántica de operación (ej. POST que antes creaba 1 recurso ahora crea múltiples).
- Eliminar valor de enum.

Proceso:

- Crear nueva versión de API (ej. v2).
- Publicar especificación OpenAPI de v2 en portal de desarrolladores.
- Desplegar v2 en paralelo con v1 (ambas coexisten).
- Notificar a clientes con mínimo 3 meses de antelación: "v1 será deprecated en YYYY-MM-DD, migrar a v2".
- Periodo de transición: 6 meses donde v1 marca como deprecated pero funciona.
- Tras 6 meses, eliminar v1 (retorna HTTP 410 Gone).

8.3.2. Pipeline de despliegue (CI/CD)

Fases:

1. **Commit** → Desarrollador hace commit a repositorio Git (GitLab, GitHub).
2. **Build** → CI (GitLab CI, Jenkins) compila código, ejecuta tests unitarios.
3. **Test** → Tests de integración en entorno de test (contenedores efímeros).
4. **Security scan** → Escaneo de vulnerabilidades (SonarQube, Snyk), análisis de dependencias.
5. **Package** → Construcción de imagen Docker, tag con versión + commit SHA.
6. **Deploy to PRE** → Despliegue automático a entorno de preproducción (Kubernetes namespace thot-pre).
7. **Smoke tests in PRE** → Tests básicos de humo (¿APIs responden? ¿BBDD accesible?).
8. **Manual approval** → Aprobación humana para desplegar a producción (solo para cambios en APIs críticas).
9. **Deploy to PROD** → Despliegue a producción con estrategia **rolling update** (Kubernetes despliega nuevas réplicas gradualmente, verifica health checks, si falla rollback automático).
10. **Post-deploy monitoring** → Monitorización intensiva durante 1 hora post-despliegue (alertas más sensibles), equipo de guardia atento.

Frecuencia de despliegues:

- **Preproducción**: Múltiples veces al día (cada merge a rama main).

- **Producción**: 1-2 veces por semana (martes/jueves, evitar viernes/fines de semana).

Rollback:

- Si despliegue causa incremento de errores o latencia, rollback automático (Kubernetes mantiene versión anterior, revertir en <5 minutos).
- Si problema se detecta horas después, rollback manual (desplegar versión anterior explícitamente).

8.4. Pruebas de integración y regresión

(plan de test, entornos de prueba, datos de prueba)

9. SINCRONIZACIÓN Y OPERACIÓN EN TIEMPO REAL/OFFLINE

fujos que requieren inmediatez vs. toleran retraso, reglas de sincronización (qué/cuándo/prioridad/granularidad) y consistencia esperada. Describe el funcionamiento offline (Lote 2) (almacenamiento local, cola y resolución de conflictos al reconectar)

Esta sección detalla la arquitectura de soporte para operación en campo sin conectividad, sincronización diferida y resolución de conflictos.

9.1. Arquitectura de operación desconectada

9.1.1. Capacidades offline en dispositivos Lote 2

Datos almacenables offline (en BBDD local SQLite cifrada):

- **Catálogos maestros** (sincronizados periódicamente desde central):
  - Tipos de vestigios (lista con descripciones/iconos).
  - Plantillas de inspección (flujos de trabajo predefinidos).
  - Datos de referencia (ej. catálogo de sustancias químicas peligrosas).
- **Casos asignados** al agente (metadatos de asuntos, vestigios ya registrados, pueden consultarse offline).
- **Cola de operaciones pendientes** (operaciones de escritura hechas offline, esperando sincronización).

Operaciones soportadas offline:

- **Registrar vestigio** (con archivos adjuntos): Se almacena en BBDD local + object storage local (directorio cifrado del dispositivo), se marca como "pendiente de sincronización".
- **Actualizar vestigio** (ej. añadir observaciones): Se almacena modificación como "delta" en cola.
- **Registrar evento de cadena de custodia**: Se encola evento para sincronizar.
- **Consultar datos locales** (casos/vestigios previamente descargados): Lectura directa de BBDD local, instantánea.

Operaciones NO soportadas offline (requieren conexión):

- Consultas biométricas (requieren acceso a ABIS remoto).
- Consultas a sistemas externos (PERSONAS, INTERPOL).
- Asistencia remota WebRTC (requiere streaming tiempo real).

9.1.2. Detección de conectividad

App Lote 2 monitoriza estado de red continuamente:

Estados de conectividad:

1. **Online** (verde): Conexión activa a THOT (verificado con ping cada 30 segundos a endpoint /api/v1/health).

2. **Offline** (rojo): Sin conexión (timeout de ping, o red deshabilitada).
3. **Degradada** (amarillo): Conexión intermitente o muy lenta (ping tarda >5 segundos, o tasa de error >20%).

Indicador visual en UI: Icono en barra de estado (verde/amarillo/rojo) + texto descriptivo ("Conectado", "Sin conexión - datos se sincronizarán al reconectar", "Conexión lenta").

Comportamiento adaptativo:

- Si **online**: Operaciones de escritura se envían inmediatamente a THOT, lecturas se hacen contra API (con fallback a caché local si falla).
- Si **offline**: Operaciones de escritura se encolan localmente, lecturas se hacen contra BBDD local, UI muestra banner "Modo offline".
- Si **degradada**: App pregunta al usuario: "Conexión lenta detectada. ¿Enviar datos ahora o esperar a mejor conexión?" (usuario decide si urgencia justifica usar conexión lenta).

9.2. Colas persistentes y sincronización diferida

9.2.1. Arquitectura de cola offline

Implementación: SQLite con tabla offline_queue:

```
CREATE TABLE offline_queue (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  event_id TEXT UNIQUE NOT NULL,
  event_type TEXT NOT NULL, -- 'vestigio.creado', 'coc.evento', etc.
  timestamp TEXT NOT NULL,  -- ISO 8601 UTC
  payload TEXT NOT NULL,    -- JSON serializado
  attachments TEXT,         -- Lista de paths locales de archivos adjuntos
  checksum TEXT NOT NULL,   -- SHA-256 del payload
  retry_count INTEGER DEFAULT 0,
  status TEXT DEFAULT 'pending',  -- 'pending', 'syncing', 'synced', 'failed'
  error_message TEXT,
  created_at TEXT DEFAULT CURRENT_TIMESTAMP
);

CREATE INDEX idx_status ON offline_queue(status);
CREATE INDEX idx_timestamp ON offline_queue(timestamp);
```

Flujo de encolado:

1. Usuario en app (offline) registra vestigio.
2. App valida datos localmente (JSON Schema).
3. App calcula checksum de payload + archivos.
4. App inserta registro en tabla offline_queue (status=pending).
5. App almacena archivos adjuntos en directorio local (/storage/offline_attachments/{event_id}/).
6. App muestra confirmación al usuario: "Vestigio registrado localmente, se sincronizará al conectar" (icono de nube con flecha arriba).

9.2.2. Proceso de sincronización

Trigger de sincronización:

- **Automático**: Al detectar reconexión (listener de red dispara sincronización).
- **Manual**: Usuario presiona botón "Sincronizar ahora" en UI.
- **Programado**: Cada 6 horas si hay conexión (sincroniza cambios maestros desde central incluso si no hay eventos pendientes).

Algoritmo de sincronización:

1. App consulta tabla offline_queue WHERE status IN ('pending', 'failed' AND retry_count <3).
2. Ordena eventos por timestamp ascendente (mantener orden cronológico original).
3. Para cada evento: a. Cambia status a syncing. b. Prepara payload: comprime archivos adjuntos (gzip), construye multipart request con metadata + archivos. c. Envía POST a /api/v1/sync/upload con batch de eventos (máximo 50 eventos por batch para no saturar). d. THOT responde con resultado por evento (success, failure + razón). e. Para eventos exitosos: cambiar status a synced, eliminar archivos locales adjuntos (liberar espacio). f. Para eventos fallidos: incrementar retry_count, cambiar status a failed, almacenar error_message, programar reintento con backoff exponencial.
4. Al completar batch, mostrar notificación al usuario: "Sincronizados 45 eventos, 2 pendientes por error" (usuario puede ver detalles en log de sincronización).

Política de reintentos:

- **Errores transitorios** (timeout, HTTP 5xx): Reintentar con backoff exponencial (1min, 2min, 4min), máximo 3 intentos, luego marcar como fallido permanentemente y notificar.
- **Errores permanentes** (HTTP 400 validación, HTTP 409 conflicto): No reintentar, marcar como fallido, mostrar mensaje específico al usuario para corrección manual.

9.3. Resolución de conflictos

Escenario de conflicto: Vestigio modificado en dispositivo de campo (offline) y en THOT central durante periodo de desconexión. Al sincronizar, THOT detecta que versión central ha cambiado desde que dispositivo descargó originalmente.

Cofinanciado por la Unión Europea
Fondos Europeos

9.3.1. Estrategias de resolución

9.3.1.1. Estrategia 1: Last-Writer-Wins (LWW)

Definición: La modificación con timestamp más reciente prevalece, descartando modificación anterior.

Aplicación: Usada por defecto para campos de metadata descriptiva (ej. descripcion, observaciones), donde no hay "verdad absoluta" y se asume que modificación más reciente refleja información más actualizada.

Implementación:

- Dispositivo envía modificación con timestamp 2026-01-21T10:00:00Z.
- THOT verifica: última modificación en central fue 2026-01-21T09:00:00Z.
- Timestamp de dispositivo es posterior → se acepta modificación de dispositivo, sobreescribe central.

Limitación: Si relojes no están perfectamente sincronizados (NTP drift), puede haber decisión incorrecta. Mitigación: Dispositivos sincronizan reloj con NTP frecuentemente, THOT verifica que timestamp de evento no está "demasiado en futuro" (>5 minutos adelantado = sospechoso, se rechaza).

9.3.1.2. Estrategia 2: Merge automático (3-way merge)

Definición: Intentar fusionar cambios de ambas versiones si afectan campos diferentes.

Aplicación: Usada para objetos complejos con múltiples campos independientes (ej. vestigio con campos descripcion, observaciones, etiquetas).

Implementación:

- **Base**: Versión original descargada por dispositivo (checksum conocido).
- **Central**: Versión actual en THOT (campos A y B modificados).
- **Campo**: Versión modificada en dispositivo (campo C modificado).
- **Merge**: Si modificaciones no solapan (A/B vs C), fusionar: tomar cambios de central en A/B, cambios de dispositivo en C, generar versión fusionada.
- Si modificaciones solapan (mismo campo modificado en ambos), fallar a estrategia LWW o escalar a revisión manual.

9.3.1.3. Estrategia 3: Revisión manual

Definición: Conflicto se marca para revisión humana, usuario experto decide qué versión prevalece o cómo fusionar.

Aplicación: Usada para campos críticos forenses (ej. checksums de archivos, resultados de análisis) donde modificación automática no es aceptable.

Implementación:

Cofinanciado por la Unión Europea
Fondos Europeos

- THOT detecta conflicto en campo crítico.
- Crea ticket en sistema de gestión de conflictos (UI web).
- Notifica a supervisor del dispositivo: "Conflicto requiere revisión en vestigio VES-2026-00001234".
- Supervisor accede a UI, ve versión de campo (central) y versión de dispositivo (lado a lado).
- Supervisor elige manualmente: aceptar central, aceptar campo, o crear versión fusionada manualmente.
- Decisión se registra en log de auditoría (quién decidió qué, cuándo).

9.3.2. *Prevención de conflictos*

Bloqueo optimista (Optimistic Locking): Cada registro tiene campo `version` (número entero incrementado en cada modificación) o `last_modified_at` (timestamp). Al modificar, cliente envía versión que conoce, THOT verifica que no ha cambiado, si cambió rechaza con HTTP 409 Conflict + versión actual.

Bloqueo pesimista (Pessimistic Locking) (no usado típicamente en THOT en ausencia de conectividad con el Lote 2, pero mencionado para completitud): Cliente "lockea" recurso antes de modificarlo (`POST /api/v1/vestigios/{id}/lock`), modifica, libera lock. Otros clientes no pueden modificar mientras está lockeado. **Problema**: Incompatible con operación offline (cliente offline no puede lockear).

9.4. Políticas de Quality of Service (QoS) aplicativo

Cuando sincronización ocurre con conectividad limitada (ej. 3G lento, ancho de banda <1 Mbps), es necesario priorizar qué datos sincronizar primero.

9.4.1. *Niveles de prioridad de eventos*

Prioridad	Tipo de evento	Justificación	Ejemplo
Crítica	Eventos de cadena de custodia, vestigios con match biométrico	Urgencia operativa/legal	Registro de vestigio con huella que matchea con sospechoso conocido
Alta	Vestigios con archivos pequeños (<1 MB)	Sincronización rápida posible	Fotografía de huella dactilar
Media	Vestigios con archivos medianos (1-10 MB)	Importante pero no urgente	Vídeo corto de inspección
Baja	Vestigios con archivos grandes (>10 MB), telemetría histórica	Puede esperar a mejor conexión	Vídeo 4K de escena completa, logs de GPS históricos

Tabla 25. Niveles de prioridad de eventos en THOT para la gestión y sincronización de información, estableciendo criterios de urgencia operativa y legal (Cadena de Custodia y coincidencias biométricas) y priorización por tamaño/criticidad del contenido, con ejemplos representativos para guiar el tratamiento y el orden de procesamiento.

9.4.2. Algoritmo de sincronización con QoS

1. Al iniciar sincronización, medir ancho de banda disponible (enviar paquete de prueba pequeño, medir tiempo).
2. Clasificar eventos en cola según prioridad.
3. Si ancho de banda alto (>5 Mbps): Sincronizar todos los eventos en orden cronológico.
4. Si ancho de banda medio (1-5 Mbps):
    - Sincronizar eventos críticos/altos primero (todos).
    - Sincronizar eventos medios si tiempo disponible.
    - Diferir eventos bajos hasta próxima sincronización con mejor conexión.
5. Si ancho de banda bajo (<1 Mbps):
    - Sincronizar solo eventos críticos (metadata + archivos pequeños).
    - Diferir resto.
    - Comprimir archivos agresivamente (ej. reducir resolución de imágenes temporalmente, enviar versión reducida, marcar para "sincronización completa posterior").

Configuración por usuario: App permite a supervisor configurar política de QoS (ej. "Siempre sincronizar todo", "Modo ahorro de datos: solo crítico", "Automático: decidir según conexión").

Cofinanciado por la Unión Europea
Fondos Europeos

10. PLAN DE TRANSICIÓN Y MIGRACIÓN

Esta sección detalla cómo se desplegará la arquitectura de interoperabilidad en entorno de producción, gestionando coexistencia con sistemas actuales y minimizando disrupción operativa.

10.1. Estrategia de coexistencia entre sistemas legacy y nuevos

10.1.1. Fase 0: Preparación (pre-despliegue, duración: 2 meses)

Actividades:

1. **Auditoría de entorno**: Evaluar infraestructura actual de Policía Nacional (servidores disponibles, capacidades de red, sistemas legacy existentes), identificar dependencias y restricciones.
2. **Provisión de infraestructura**: Preparar cluster Kubernetes (instalación, configuración de red, storage), provisión de servidores físicos/VMs si necesario.
3. **Instalación de componentes base**: Desplegar Kafka, PostgreSQL, MongoDB, Redis, MinIO, Hyperledger Fabric en cluster, configurar replicación y backups.
4. **Configuración de seguridad**: Emitir certificados mTLS (CA de THOT), configurar firewall, integrar con IdP corporativo (OAuth 2.0), configurar Vault para gestión de secretos.
5. **Carga de datos maestros iniciales**: Migrar catálogos existentes (tipos de vestigios, plantillas) a THOT.

Criterios de salida: Infraestructura lista, componentes base desplegados y verificados (smoke tests), acceso configurado.

10.1.2. Fase I: Piloto controlado (duración: 3 meses)

Alcance: Despliegue en **2-3 unidades territoriales piloto** (ej. Madrid, Barcelona), con **10-20 agentes** seleccionados.

Funcionalidades habilitadas:

- Registro de vestigios desde campo (dispositivos Lote 2) con sincronización online/offline.
- Consulta de vestigios desde central (UI web).
- Cadena de custodia básica (registro en registro inmutable).
- Integración con 1 sistema externo (ej. ABIS para cotejos biométricos).

Metodología:

- **Entrenamiento**: Sesiones presenciales (2 días) + documentación/vídeos para agentes piloto y supervisores.
- **Soporte intensivo**: Equipo de THOT disponible 24/7 (guardias), respuesta a incidencias <1 hora.
- **Recolección de feedback**: Encuestas semanales, entrevistas con usuarios, análisis de logs de uso.
- **Métricas objetivo**:
  - Disponibilidad del sistema >99% (downtime <7 horas/mes).
  - Latencia p95 de APIs <3 segundos.

- Tasa de adopción: >80% de agentes piloto usan THOT para registrar vestigios (vs sistema legacy).
- Satisfacción de usuarios: puntuación >7/10 en encuestas.

Criterios de éxito para pasar a Fase II:

- Todas las métricas objetivo alcanzadas.
- 0 incidencias críticas sin resolver.
- Feedback mayoritariamente positivo (>70% usuarios satisfechos).
- Aprobación formal de Policía Nacional (comité de proyecto).

10.1.3. Fase II: Expansión gradual (duración: 6 meses)

Alcance: Expansión a **20 unidades territoriales adicionales** (total ~25 de ~100), incorporando **200-300 agentes**.

Funcionalidades adicionales habilitadas:

- Asistencia remota WebRTC.
- Integraciones con sistemas internos adicionales (PERSONAS, sistemas judiciales).
- Portal de análisis avanzado (dashboards para investigadores).

Metodología:

- **Despliegue escalonado**: Activar 4-5 unidades por mes (evitar saturar soporte).
- **Entrenamiento escalable**: Formación de "champions" locales (1-2 por unidad) que entrenan a resto de agentes, sesiones virtuales (webinars).
- **Soporte distribuido**: Equipo de soporte ampliado, tiempos de respuesta estándar (crítica <4h, alta <24h).
- **Optimización continua**: Aplicar lecciones aprendidas de Fase I, ajustar configuración (ej. políticas de caché, umbrales de alertas) basándose en telemetría real.

Métricas objetivo (actualizadas):

- Disponibilidad >99.5%.
- Latencia p95 <2 segundos.
- Tasa de incidencias críticas <1/semana.
- Tasa de adopción >90%.

10.1.4. Fase III: Despliegue completo (duración: 12 meses)

Alcance: Despliegue en **todas las unidades territoriales** (~100), todos los agentes (~1000-2000).

Funcionalidades completas:

- Todas las funcionalidades planificadas habilitadas.
- Integraciones internacionales (INTERPOL, EUROPOL) activadas.
- Analíticas avanzadas y ML/AI integrados.

Cofinanciado por la Unión Europea
Fondos Europeos

Metodología:

- **Despliegue por oleadas**: 10 unidades/mes durante 8 meses, últimas unidades pequeñas/remotas.
- **Automatización de operaciones**: Monitorización y gestión de incidencias completamente operativa, runbooks para problemas comunes, escalado automático bien calibrado.
- **Transición a BAU (Business As Usual)**: Transferencia de responsabilidad operativa de equipo de proyecto a equipo de operaciones de Policía Nacional (si existe) o equipo de mantenimiento del contratista.

Hitos clave:

- **Mes 6**: 50% de unidades desplegadas.
- **Mes 10**: 90% de unidades desplegadas.
- **Mes 12**: 100% desplegado, sistema legacy descomisionado (si aplicable).

10.2. Fases de despliegue de integraciones

Durante fases I-III, THOT coexistirá con sistemas legacy existentes. Es crítico gestionar esta coexistencia para evitar duplicación de trabajo y confusión.

10.2.1. Estrategias de coexistencia

10.2.1.1. Opción A: "Dual entry" (entrada dual)

Descripción: Agentes registran evidencias en THOT **y** en sistema legacy (temporalmente), hasta que THOT se valide completamente.

Ventajas: Máxima seguridad (si THOT falla, datos están en legacy), permite comparar resultados.

Desventajas: Doble trabajo para agentes (frustración, baja adopción), riesgo de inconsistencias (olvido de registrar en uno de los sistemas).

Cuándo usar: Fase I piloto (solo primeras 2-4 semanas), para validar que THOT registra datos correctamente comparando con legacy.

10.2.1.2. Opción B: "Cutover progresivo"

Descripción: En cada unidad desplegada, agentes dejan de usar sistema legacy completamente y pasan a THOT (no coexistencia en misma unidad, pero legacy sigue activo en unidades no desplegadas).

Ventajas: Evita doble trabajo, simplifica operaciones en unidad desplegada.

Desventajas: Requiere migración de datos históricos si agentes necesitan acceder a casos antiguos (ver sección 10.3), legacy debe mantenerse operativo hasta que última unidad migre.

Cuándo usar: Fases II-III, opción recomendada.

10.2.1.3. Opción C: "Legacy read-only"

Descripción: Al desplegar THOT en unidad, sistema legacy se pone en modo "solo lectura" para esa unidad (agentes pueden consultar datos antiguos pero no crear nuevos registros).

Ventajas: Acceso a histórico garantizado sin migración masiva, forzar adopción de THOT.

Desventajas: Requiere modificar sistema legacy (si es posible técnicamente), dos interfaces para agentes (THOT para nuevo, legacy para histórico).

Cuándo usar: Si migración de datos históricos no es viable técnicamente, opción de fallback.

10.2.2. Integración con sistemas legacy

Si sistemas legacy exponen APIs (ej. SOAP, REST), crear **adaptadores bidireccionales**:

THOT → Legacy: Enviar copia de datos registrados en THOT a sistema legacy (para unidades que aún lo requieren).

Legacy → THOT: Consultar datos históricos de legacy desde THOT (si agente necesita acceder a caso antiguo, THOT consulta legacy transparentemente).

10.3. Plan de migración de datos históricos

10.3.1. Evaluación de necesidad de migración

Preguntas clave:

- ¿Qué antigüedad tienen datos en legacy que agentes necesitan acceder regularmente? (ej. casos de últimos 2 años = alta prioridad, >5 años = baja prioridad).
- ¿Volumen de datos? (GB, número de registros).
- ¿Calidad de datos en legacy? (¿Están estructurados, limpios, o requieren limpieza masiva?).

Decisión:

- **Migración completa** (si volumen manejable <1TB, datos estructurados, alta necesidad de acceso): Migrar todos los datos históricos a THOT antes de Fase I.
- **Migración selectiva** (si volumen grande o calidad baja): Migrar solo casos activos (no cerrados >5 años), resto accesible vía integración Legacy→THOT.
- **Sin migración** (si legacy es muy antiguo, datos no estructurados): Mantener legacy read-only, no migrar, descomentar tras N años según política de retención.

10.3.2. Proceso de migración (si se decide migrar)

1. **Extracción**: Exportar datos de legacy (SQL dump, API calls, scraping si no hay API).

2. **Transformación**: Mapear esquema de legacy a esquema canónico THOT (scripts ETL en Python/Spark).
    - Normalizar formatos (fechas, coordenadas GPS).
    - Validar integridad (checksums de archivos, referencias entre tablas).
    - Limpiar datos (eliminar duplicados, corregir valores erróneos si posible).
3. **Carga**: Importar datos a THOT (bulk insert en PostgreSQL, archivos a MinIO), validar que importación fue exitosa (conteo de registros, muestreo).
4. **Verificación**: Comparar muestra aleatoria (ej. 1000 registros) entre legacy y THOT, verificar que datos coinciden.
5. **Reconciliación**: Corregir discrepancias encontradas en verificación.

Timeline estimado: 2-4 semanas para dataset de tamaño medio (100K registros, 100 GB archivos).

10.4. Plan de reversión y contingencias

Si durante despliegue ocurre problema crítico que hace THOT inusable, debe existir plan de **rollback** (vuelta atrás) rápido.

10.4.1. Niveles de rollback

Rollback técnico (reversión de código/config):

- **Cuándo**: Despliegue de nueva versión causa errores críticos (HTTP 500, crashs).
- **Cómo**: Kubernetes rollback a versión anterior de imagen Docker (comando `kubectl rollout undo deployment/thot-api`), reversión automática <5 minutos.

Rollback funcional (revertir a sistema legacy):

- **Cuándo**: Problema sistémico de THOT que no puede resolverse rápidamente (ej. corrupción de datos, fallo de diseño detectado).
- **Cómo**:
    1. Poner THOT en modo "solo lectura" (deshabilitar endpoints de escritura).
    2. Reactivar sistema legacy en modo escritura (si estaba read-only).
    3. Notificar a agentes vía email/SMS: "Revertir temporalmente a sistema legacy hasta nuevo aviso".
    4. Sincronizar datos creados en THOT durante periodo activo hacia legacy (si adaptador bidireccional existe).
    5. Analizar causa raíz del problema, implementar fix, probar exhaustivamente, re-desplegar THOT.

Criterios para decidir rollback:

- Disponibilidad <90% durante >4 horas.
- Incidencia crítica que impide registro de evidencias sin workaround.
- Pérdida de datos detectada (corruption, fallos de backup).
- Decisión de Policía Nacional (usuario final) por razones operativas.

10.4.2. Comunicación en caso de rollback

Inmediata (durante evento):

- Email urgente a todos los usuarios afectados: "THOT temporalmente no disponible, usar sistema legacy. Notificaremos al resolver."
- Actualizar status page (https://status.thot.policia.es): "Major outage - Investigating".

Post-mortem (tras resolución):

- Informe detallado de incidencia: qué ocurrió, por qué, cómo se resolvió, acciones correctivas.
- Presentación a comité de proyecto y usuarios.
- Documentación de lecciones aprendidas.

10.5. Impacto en procesos y formación de usuarios

10.5.1. Audiencias de formación

Audiencia	Rol	Necesidades de formación
Agentes de campo (Técnicos Policía Científica)	Usuarios primarios de app Lote 2	Uso de app para registrar vestigios, sincronización offline, procedimientos CoC
Analistas centrales	Usuarios de UI web THOT	Consulta de vestigios, solicitud de análisis, interpretación de resultados
Investigadores	Usuarios de portal de análisis	Dashboards, búsquedas avanzadas, generación de informes
Supervisores/Coordinadores	Usuarios de funciones de gestión	Asignación de casos, revisión de CoC, monitorización de operaciones
Administradores de sistemas	Personal IT de Policía Nacional	Administración de THOT (gestión de usuarios, monitorización, troubleshooting)
Desarrolladores externos (Lote 2, sistemas TIC)	Integradores	APIs de THOT, SDK, documentación técnica, testing

Tabla 26. Audiencias objetivo del plan de formación de THOT y su correspondencia con roles operativos y técnicos, detallando las necesidades formativas por perfil (captura en campo, análisis y consulta, supervisión, administración de sistemas e integración/desarrollo) para asegurar adopción, uso correcto y cumplimiento de procedimientos (incluida Cadena de Custodia).

10.5.2. Modalidades de formación

Presencial:

- **Duración**: 1-2 días según audiencia.
- **Contenido**: Teoría (presentación de arquitectura, funcionalidades) + práctica (hands-on en entorno de training).
- **Público**: Fase I piloto (formación intensiva), formadores de formadores (champions).

Virtual (webinars, e-learning):

- **Duración**: Módulos de 30-60 minutos.
- **Contenido**: Vídeos demostrativos, tutoriales interactivos, quizzes.
- **Público**: Fases II-III (escalable a gran número de usuarios).

Documentación:

- **Manuales de usuario** (PDF/HTML): Guías paso a paso con screenshots.
- **FAQs**: Preguntas frecuentes con respuestas.
- **Vídeos tutoriales** (YouTube privado o plataforma LMS): Screencast de operaciones comunes (5-10 minutos por vídeo).

Soporte continuo:

- **Helpdesk**: Email/teléfono para consultas (SLA: respuesta <24h).
- **Foro comunitario** (interno Policía Nacional): Usuarios ayudan a usuarios, moderado por equipo THOT.
- **Sesiones de "office hours"**: 1 hora semanal donde usuarios pueden conectarse por videoconferencia y hacer preguntas en vivo.

10.5.3. Métricas de adopción y uso

Para medir éxito de formación y gestión del cambio, monitorizar:

Métrica	Definición	Objetivo	Fuente de datos
Tasa de adopción	% de usuarios activos / usuarios totales desplegados	>90% en 1 mes post-formación	Logs de autenticación (usuarios únicos/día)
Frecuencia de uso	Promedio de operaciones por usuario/día	>5 operaciones/día (agentes campo)	Logs de API (conteo de POST/PATCH por userId)
Satisfacción de usuarios	Puntuación en encuestas (1-10)	>7/10	Encuestas mensuales (Google Forms, integrada en UI)

Métrica	Definición	Objetivo	Fuente de datos
Tickets de soporte	Número de tickets por 100 usuarios	<5 tickets/100 usuarios/mes	Sistema de ticketing (Jira, ServiceNow)
Tiempo de resolución de tickets	Mediana de tiempo de cierre	<48 horas	Sistema de ticketing

Tabla 27. Métricas de adopción y uso para evaluar la eficacia de la formación y la gestión del cambio en THOT, definiendo objetivos medibles y fuentes de datos (logs de autenticación y APIs, encuestas y sistema de ticketing) para monitorizar adopción, frecuencia de uso, satisfacción y carga/eficiencia del soporte.

Acciones correctivas si métricas no se alcanzan:

- Tasa de adopción baja (<70%) → Revisar usabilidad de UI, sesiones de formación adicionales, incentivos para uso.
- Satisfacción baja (<6/10) → Focus groups con usuarios insatisfechos, identificar pain points, priorizar mejoras.
- Tickets altos → Mejorar documentación, añadir FAQs, videos tutoriales de problemas recurrentes.

10.6. Criterios de aceptación por fase

11. ANEXOS

Definiciones y material de referencia

Cofinanciado por la Unión Europea
Fondos Europeos