UTE FORENSIA THOT

F1.1.4 Plan de Pruebas

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 Plan de Pruebas
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:	Alta
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".

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

TABLA DE CONTENIDOS

1. INTRODUCCIÓN

1.1. Resumen Ejecutivo

El presente *F1.1.4 Plan de Pruebas* define el enfoque integral para la verificación y validación del sistema THOT durante la Fase 2 del proyecto, garantizando que el desarrollo cumpla los requisitos funcionales y técnicos establecidos en el pliego y la memoria técnica. El plan articula una estrategia de pruebas basada en metodología ágil, alineando épicas, historias de usuario y casos de prueba con los entregables previstos, y asegurando una cobertura completa mediante pruebas unitarias, de integración, funcionales, no funcionales, interoperabilidad, ciberseguridad, continuidad y restauración de datos.

La estructura del plan prioriza la trazabilidad entre requisitos, desarrollos y evidencias de prueba, apoyándose en prácticas de automatización en CI/CD y herramientas especializadas para garantizar calidad, seguridad y rendimiento. Se definen métricas clave de control, procedimientos de comunicación, mecanismos de gestión de riesgos y criterios de aceptación orientados a asegurar un incremento continuo de valor en cada sprint, minimizando la deuda técnica y asegurando la robustez del sistema en escenarios críticos.

Finalmente, el plan establece un marco de coordinación entre los distintos socios de la UTE, desarrolladores, especialistas QA, expertos en ciberseguridad y representantes del cliente, asegurando la correcta planificación, seguimiento y ejecución de las actividades de prueba. Con ello, se proporciona la base necesaria para validar el prototipo del sistema THOT y preparar su transición hacia la fase pre-operacional, garantizando altos estándares de seguridad, interoperabilidad y fiabilidad.

1.2. Propósito del documento

El propósito del documento *F1.1.4 Plan de Pruebas* consiste en establecer de manera formal y detallada la planificación de las actividades de verificación y validación que se llevarán a cabo durante la Fase 2: Desarrollo de un prototipo y pruebas de la solución propuesta.

Como entregable correspondiente a la Fase I (Diseño de la solución) dentro del Lote 1 (Plataforma Interoperable de Servicios de Inteligencia Forense con Soluciones Innovadoras de Apoyo a la Gestión del Servicio), este plan tiene como objetivo fundamental asegurar la completitud y cobertura de las pruebas necesarias para garantizar que la solución desarrollada se adecúe a los requisitos funcionales y técnicos definidos en el pliego.

1.3. Alcance

El objetivo de este documento es proporcionar la información y el marco necesarios para planificar y ejecutar todos los procesos de prueba requeridos para la validación de la interoperabilidad de la plataforma THOT.

1.4. Referencias

Los siguientes documentos describen el diseño de la solución THOT, y constituyen la base de referencia para la elaboración y ejecución del presente Plan de Pruebas:

- F1.1.1. Documento de prestaciones técnicas y funcionales
- F1.1.2. Documento de arquitectura del sistema
- F1.1.3. Documento de modelo de datos

1.5. Glosario

Término	Descripción
Cadena de Custodia	Proceso que garantiza la integridad y trazabilidad de la evidencia digital.
Caso de Prueba	Secuencia definida para validar una funcionalidad mediante entradas y resultados esperados.
CI/CD	Proceso automatizado de integración y entrega continua de software.
Criterios de Aceptación	Condiciones que deben cumplirse para validar correctamente una historia de usuario.
Épica	Conjunto de historias de usuario relacionadas que estructuran el desarrollo.
Historia de Usuario	Unidad funcional que describe una necesidad desde la perspectiva del usuario.
Interoperabilidad	Capacidad de distintos sistemas para comunicarse y trabajar juntos.
Ledger	Libro de registros inmutable utilizado para garantizar trazabilidad y no repudio.
RPO	Antigüedad máxima de los datos que puede tolerarse tras una restauración.
RTO	Tiempo máximo permitido para restaurar un servicio tras un fallo.
SLO	Objetivo cuantificable de nivel de servicio en rendimiento y disponibilidad.

2. COMPLETITUD Y COBERTURA DEL PLAN DE PRUEBAS

A continuación, se detallan los aspectos más relevantes del contexto al que se aplicará el plan de pruebas, incluyendo a qué elementos aplica, el alcance previsto para las mismas y el detalle de actores interesados y su papel en el desarrollo y aplicación del plan de pruebas.

2.1. Proyecto de pruebas

El presente Plan de Pruebas aplicará al sistema THOT en su extensión completa. Este plan de pruebas se complementa con el F1.3.3. Plan de Pruebas de Interoperabilidad, enfocado principalmente a la validación de las interacciones del sistema THOT con sistemas de terceros (lote 2 de la oferta, así como sistemas a integrar dentro de Policía Científica).

Los documentos que detallan el alcance del sistema THOT son los siguientes:

- F1.1.1. Documento de prestaciones técnicas y funcionales
- F1.1.2. Documento de arquitectura del sistema
- F1.1.3. Documento de modelo de datos

2.2. Elementos de prueba

Los elementos de prueba comprenden los componentes que integran la solución integral THOT. La arquitectura del sistema, basada en el paradigma de microservicios, propone un modelo en el que componentes con funcionalidades específicas se orquestan e interconectan para proporcionar, de manera conjunta, un servicio de alto valor destinado a la Policía Científica.

El entregable F1.1.2, Documento de Arquitectura del Sistema, describe detalladamente el diseño de dicha solución, estructurando los microservicios en categorías funcionales (servicios). Estas categorías constituyen el eje sobre el cual se articula tanto el desarrollo de la solución como la planificación de las actividades de prueba:

- Servicio Espacio de Datos
- Servicio de Procesamiento de Datos
- Servicio de Inteligencia Artificial
- Servicios de Coordinación y Gestión
- Servicio de Alertas
- Servicio de Cadena de Custodia
- Servicio de Comunicaciones

2.3. Alcance de las pruebas

El sistema THOT consiste en un desarrollo integral, diseñado específicamente para responder a las necesidades planteadas en el Reto Forense, si bien incorporará ciertos componentes preexistentes, propietarios o de código abierto bajo licencias específicas. El plan de pruebas abarca el alcance sobre los nuevos componentes, cualquier extensión funcional aplicada a los componentes ya existentes, así como la validación de la integración completa del conjunto de componentes.

2.4. Interesados

El plan de pruebas comprende el ciclo de desarrollo y entrega de los desarrollos de manera integral. La siguiente tabla detalla los diferentes actores y su papel en el desarrollo de las pruebas. Los interesados (stakeholders) incluyen tanto al personal interno del proyecto como a representantes externos. Cada uno contribuye de forma distinta al aseguramiento de la calidad del producto mediante la planificación, revisión, aprobación o ejecución de las pruebas.

Rol	Organización	Responsabilidades principales	Nivel de implicación	Frecuencia de interacción
Responsable de Pruebas	HI IBERIA	Planificar, coordinar y supervisar todas las actividades de prueba.	Alta	Diaria
Equipo de Testing	HI IBERIA, ETRA, HERTA, HARDLINK, UPV	Diseñar, ejecutar y documentar los casos de prueba.	Alta	Diaria

Cofinanciado por la Unión Europea
Fondos Europeos

Desarrolladores	HI IBERIA, ETRA, HERTA, UPV	Atender incidencias, corregir defectos y revisar resultados de pruebas.	Media	Diaria
Gestor de Proyecto	HI IBERIA, ETRA, HERTA, HARDLINK, UPV	Asegurar recursos, aprobar el plan de pruebas y gestionar riesgos.	Alta	Semanal
Responsable de Seguridad	HARDLINK	Supervisar pruebas relacionadas con seguridad y cumplimiento normativo.	Media	Según planificación
Cliente	Policía Científica	Aprobar los resultados de aceptación y validar la adecuación al negocio.	Baja	Mensual (tras cada entrega)
Auditor Externo	CDTI	Revisar entregables para cumplimiento de contrato.	Baja	Según planificación

3. ADECUACIÓN A LOS REQUISITOS FUNCIONALES Y TÉCNICOS DEL PLIEGO

La elaboración de escenarios y casos de prueba se llevará a cabo de manera iterativa, en consonancia con el backlog de cada sprint, evitando generar documentación exhaustiva desde el inicio para favorecer la adaptabilidad propia de los enfoques ágiles. Esta metodología se apoya en las épicas e historias de usuario planificadas, considerando que cada historia de usuario equivale a un escenario de prueba, ya que representa una funcionalidad completa y verificable dentro de la plataforma HORUS. A partir de dichas historias se definirán casos de prueba detallados basados en sus criterios de aceptación correspondientes.

Estrategia de mapeo

Cada historia de usuario considerada testable se desglosará en escenarios principales —que describen las secuencias de interacción entre el usuario y el sistema— y en casos de prueba ejecutables que incluirán precondiciones, pasos, datos de entrada, resultados esperados y postcondiciones. Este trabajo se llevará a cabo durante las sesiones de refinamiento del backlog y la planificación del sprint, integrando pruebas unitarias, de integración y de aceptación dentro de los flujos CI/CD de los entornos locales y de staging.

Ejecución y trazabilidad

Los escenarios se documentarán en la herramienta colaborativa Clickup, facilitando la trazabilidad de ejecuciones y resultados.

Épicas

La siguiente tabla recoge las épicas que estructurarán el desarrollo de la solución THOT. Estas épicas se han definido en alineación con el plan de trabajo establecido en la memoria técnica y con el alcance funcional previsto para la solución.

Épica	Paquete(s) de trabajo	Casos de uso vinculados	Objetivo de pruebas
EP-PT2.1 Identificación policial (reseña)	PT2.1 (T2.1.1–T2.1.3)	UC1, UC2	Validar creación de expediente, OCR, captura biométrica (NFIQ2/3), cumplimiento ISO/IEC 19794-2/5, cotejos 1:1 y 1:N (ABIS/AFIS, PERSONAS), alertas y judicialización.
EP-PT2.2A Inspección Técnico-Policial (ITP)	PT2.2 (T2.2.1–T2.2.4)	UC3	Verificar fijación multimodal de la escena, trazabilidad/ hashes, clasificación de vestigios (ISO 21043), teleasistencia CECOR, plan de remisión y seguimiento a laboratorio.
EP-PT2.2B Actuación en	PT2.2 (T2.2.1–T2.2.4)	UC4	Gestión integral en escenarios de víctimas múltiples: fijación multimodal, necroidentificación lofoscópica,

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

Emergencias (SVM)			etiquetado AM/PM, interoperabilidad internacional, trazabilidad completa bajo condiciones extremas.
EP-PT2.3A Ciclo de Inteligencia y Cadena de Custodia	PT2.3 (T2.3.1–T2.3.10)	UC5, UC7	Probar espacio de datos, gestión y explotación, ledger/cadena de custodia, distribución de tareas, alertas, dashboards, correlación automática en base a modus operandi y patrones delictivos.
EP-PT2.3B Gestión de Inventario, Formación y Calidad	PT2.3 (T2.3.1–T2.3.10)	UC8	Gestión de inventarios y recursos (ERP), planificación y ejecución de formación, aseguramiento de calidad con dashboards y microservicio IA para evaluación automática.
EP-PT2.4 Preparación Fase III	PT2.4 (T2.4.1–T2.4.4)	Transversal	Pruebas preliminares de escalabilidad, validación de prototipos, sostenibilidad y mantenimiento.
EP-PT2.5 Interoperabilidad	PT2.5 (T2.5.1–T2.5.3)	UC6	Validar integraciones, sincronización offline, contratos de API, interoperabilidad con ABIS/CODIS/IBIS/PRÜM/EURODAC, red degradada.
EP-PT4 Ciberseguridad y Continuidad	PT4 (T4.2–T4.5)	Transversal	Pentesting, análisis de vulnerabilidades, cumplimiento ENS/ISO 27001, restauración de backups, continuidad y recuperación ante desastres (Recovery Time Objective/Recovery Point Objective).

Historias de usuario / Escenarios de prueba

La siguiente tabla presenta el conjunto de historias de usuario que conformarán los sprints de desarrollo de la fase 2. Estas historias servirán como base para la definición de las pruebas específicas, las cuales se elaborarán al comienzo de cada sprint y permitirán verificar el cumplimiento de los criterios de aceptación.

Épica	Historia	Título	Criterios de aceptación
EP-PT2.1	ST-PT2.1-01a	Selección de formulario PNI	Sistema muestra opciones correctas (adulto/menor); selección guardada.
EP-PT2.1	ST-PT2.1-01b	Carga de PDF	Archivo validado (formato/tamaño); almacenamiento correcto.

EP-PT2.1	ST-PT2.1-01c	OCR sobre PDF	Extracción ≥98% de campos obligatorios.
EP-PT2.1	ST-PT2.1-01d	Validación de datos	Datos cumplen reglas de formato (DNI, fecha).
EP-PT2.1	ST-PT2.1-01e	Creación de expediente	Expediente con ID único y sello temporal.
EP-PT2.1	ST-PT2.1-02a	Captura lofoscópica	Registro decadactilar conforme a NFIQ2/NFIQ3.
EP-PT2.1	ST-PT2.1-02b	Normalización ISO	Formato ISO/IEC 19794-2 aplicado.
EP-PT2.2A	ST-PT2.2A-01a	Apertura de expediente ITP	Referencia única y sello temporal cualificado.
EP-PT2.2A	ST-PT2.2A-01b	Asignación de equipo	Asignación automática según carga y competencias.
EP-PT2.2A	ST-PT2.2A-02a	Fijación fotográfica	Presentación de fotos con metadatos y hash criptográfico.
EP-PT2.2A	ST-PT2.2A-02b	Fijación 3D	Presentación de modelo 3D generado y vinculado al expediente.
EP-PT2.2B	ST-PT2.2B-01a	Registro AM	Datos ante mortem capturados y validados.
EP-PT2.2B	ST-PT2.2B-01b	Registro PM	Datos post mortem capturados y vinculados.
EP-PT2.2B	ST-PT2.2B-02a	Etiquetado seguro	Registro seguro y trazable de cada vestigio.
EP-PT2.2B	ST-PT2.2B-02b	Cotejo ABIS	Cotejo automático contra ABIS completado.
EP-PT2.3A	ST-PT2.3A-01a	Creación ledger	Registro inmutable con firma digital.
EP-PT2.3A	ST-PT2.3A-01b	Correlación Modus Operandi	Algoritmo detecta patrones y genera alerta.
EP-PT2.3B	ST-PT2.3B-01a	Alta de equipo en ERP	Equipo registrado con trazabilidad completa.

EP-PT2.3B	ST-PT2.3B-01b	Alerta de stock	Sistema genera alerta por umbral crítico.
EP-PT2.3B	ST-PT2.3B-02a	Plan formativo	Plan creado y asignado a usuarios.
EP-PT2.4	ST-PT2.4-01a	Prueba de carga inicial	Escenario de prueba de carga ejecutado sin errores.
EP-PT2.4	ST-PT2.4-01b	Optimización recursos	Informe con mejoras aplicadas.
EP-PT2.5	ST-PT2.5-01a	Validación Pact	Contrato API validado sin incumplimientos.
EP-PT2.5	ST-PT2.5-01b	Prueba red degradada	Reconexión y reconciliación completadas.
EP-PT4	ST-PT4-01a	Pentesting crítico	Informe con hallazgos críticos entregado.
EP-PT4	ST-PT4-01b	Remediación vulnerabilidades	Todas las vulnerabilidades críticas cerradas.

4. COMUNICACIÓN DE LAS PRUEBAS

La comunicación entre los equipos de desarrollo, pruebas y partes interesadas se organizará conforme a los principios de transparencia y colaboración continua del marco de trabajo de proyectos Agile. Al inicio de cada sprint se celebrará una reunión conjunta entre el equipo de desarrollo y el equipo de pruebas con el objetivo de revisar las historias de usuario seleccionadas, acordar sus criterios de aceptación y definir los casos de prueba iniciales. Durante esta sesión se establecerán los escenarios funcionales prioritarios y se validará la viabilidad técnica de las actividades de verificación a ejecutar dentro del sprint.

A lo largo del ciclo de desarrollo, la coordinación se mantendrá mediante reuniones breves de seguimiento y el uso de herramientas de gestión compartidas, que permitirán al equipo de pruebas mostrar el avance de las pruebas, los defectos detectados y el estado de corrección de los mismos. La comunicación informal se complementará con procedimientos formalizados (ClickUp), donde se podrán registrar los incidentes y tareas derivadas de la ejecución de pruebas.

Al cierre de cada sprint, el equipo de pruebas generará un informe resumido que incluirá:

- Número total de pruebas ejecutadas, organizadas por tipo (unitarias, de integración, funcionales, regresión).
- Porcentaje de pruebas superadas y fallidas.
- Evolución del porcentaje de cobertura de código (code coverage).
- Número y criticidad de defectos abiertos y cerrados durante el sprint.
- Tendencias respecto al sprint anterior.

Con el fin de optimizar el tiempo de elaboración, se prevé que este informe sea generado de manera semiautomática mediante integraciones con las herramientas de CI/CD y los sistemas de seguimiento de pruebas.

Los defectos no resueltos al término del sprint serán reintroducidos en el backlog, etiquetados conforme a su prioridad y severidad, y programados para evaluación en la planificación del sprint siguiente. Asimismo, se establecerá un canal formal de comunicación destinado específicamente al cliente, en formato de formulario de reporte de defectos, a través del cual podrá notificar cualquier problema identificado en las entregas. Dicho canal formará parte del proceso oficial de gestión de incidencias y garantizará la trazabilidad de cada defecto desde su recepción hasta su resolución final.

Incident Registration Form			
Number	Sprint 10 – group z - 31		
Short Title	Measured ranges for NCS out of range		
Software product	Prototype A		
Status = Created			
Registration created by	Jon H	Date & time	1/11/11 &13:00
Comprehensive description	*NSC parameter Zstl exceeded lower bound during test 1 and a warning message was detected. Backlog (tech deficient) of change needed because issue associated with hardware-software interaction. See attachment for details on repeating.*		

Ilustración 1 Ejemplo de registro de incidente (Fuente: ISO29113-3)

Dada la envergadura del proyecto y el número de sprints planificados, se incorporarán mecanismos específicos para evitar la acumulación de defectos críticos entre iteraciones. En este contexto, el plan contempla:

- **Control de deuda técnica y defectos:** establecimiento de umbrales máximos de defectos abiertos según su severidad y prioridad, aplicando políticas de bloqueo para la incorporación de nuevas historias cuando dichos límites se superen.
- **Sprints de estabilización:** programación de iteraciones dedicadas principalmente a reducir incidencias pendientes y ejecutar pruebas no funcionales (como rendimiento y escalabilidad), minimizando el desarrollo de nuevas funcionalidades.
- **Refuerzo en la gestión del backlog:** priorización dinámica que garantice la resolución de defectos críticos antes de avanzar con nuevas capacidades, manteniendo la trazabilidad en ClickUp.
- **Integración continua con indicadores de calidad:** automatización de informes sobre cobertura, defectos abiertos y tiempos medios de resolución, proporcionando información clave para decisiones Go/No-Go en cada sprint.

Los informes generados al término de cada sprint formarán parte del anexo correspondiente al entregable *F2.1.2 Resultados de las Pruebas del Plan de Pruebas (Fase I).* Dichos informes constituirán la base sobre la cual este entregable sustentará la justificación del nivel de calidad alcanzado por el prototipo al finalizar la Fase 2.

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

5. REGISTRO DE RIESGOS

Este registro reúne los riesgos que podrían influir en la ejecución del plan de pruebas del sistema THOT. Cada riesgo se analiza según su probabilidad e impacto, y se establecen estrategias de mitigación destinadas a asegurar la adecuada cobertura, calidad y trazabilidad de las pruebas.

ID	Riesgo asociado a pruebas	Descripción	Probabilidad	Impacto	Mitigación
R1	Cobertura insuficiente por falta de personal QA	Escasez de recursos para diseñar y ejecutar pruebas, reduciendo la cobertura funcional y técnica.	Baja	Alta	Reasignar recursos, priorizar automatización, incorporar QA externos si es necesario.
R2	Documentación excesiva ralentiza pruebas	Sobrecarga en generación manual de informes que retrasa ejecución.	Media	Media	Automatizar reportes con herramientas CI/CD y plantillas predefinidas.
R3	Resultados de pruebas muestran calidad insuficiente	Al cierre del sprint, defectos críticos impiden aceptación.	Media	Alta	Integrar pruebas continuas en pipelines, definir criterios de aceptación claros y revisiones tempranas.
R4	Pruebas complejas con datos correlacionados difíciles de simular	Escenarios que requieren grandes volúmenes de datos interrelacionados (p.ej., biometría + cadena de custodia).	Alta	Alta	Generar datasets sintéticos, anonimizar datos reales cuando sea posible, usar herramientas de generación automática.
R5	Retraso en pruebas no funcionales (rendimiento, escalabilidad)	Falta de tiempo para pruebas de carga y estrés en entornos staging.	Baja	Alta	Incluir hitos específicos en cronograma, usar herramientas como k6 para automatización.

R6	Validación incompleta de ciberseguridad	Aparición de vulnerabilidades críticas al ejecutar pruebas (SAST, DAST, MITM) en componentes sensibles.	Baja	Alta	Integrar pruebas de seguridad en CI/CD, checklist ENS, auditorías periódicas.
R7	Falta de trazabilidad entre requisitos y pruebas	Riesgo de no demostrar cobertura de requerimientos.	Baja	Alta	Mantener matriz de trazabilidad en ClickUp, vincular historias de usuario con casos de prueba y métricas.

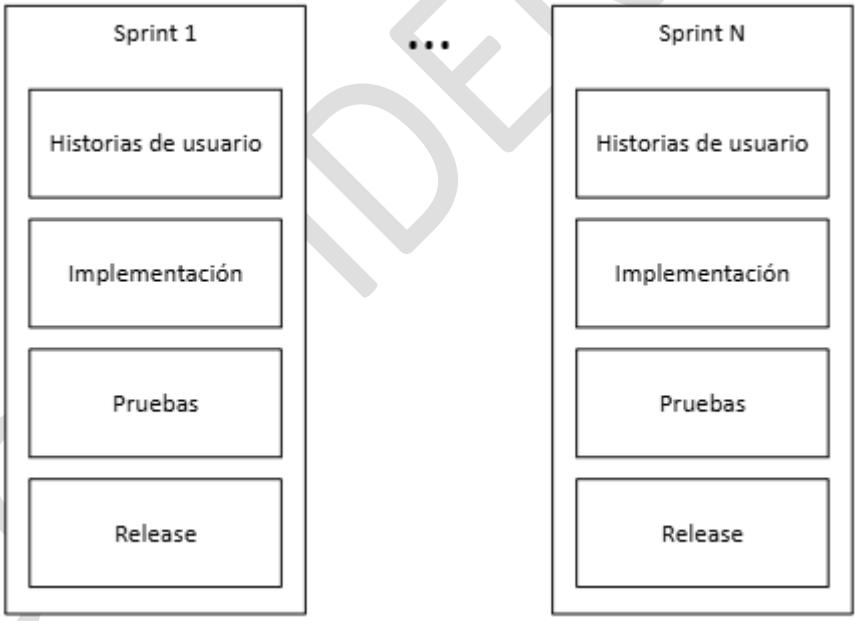
6. Definición de escenarios de prueba, casos de uso, datos de prueba, criterios de aceptación y las métricas de éxito

La estrategia de pruebas del proyecto THOT se alineará con la metodología Agile, con el fin de garantizar la calidad continua de los resultados de cada sprint, e integrando las actividades de verificación y validación como parte inherente del ciclo de desarrollo iterativo.

El equipo de desarrollo y el equipo de pruebas colaborarán estrechamente desde la planificación del sprint, definiendo criterios de aceptación claros y casos de prueba derivados de las historias de usuario incluidas en el mismo.

Las pruebas abarcarán distintos niveles: unitarias, de integración, funcionales y de aceptación. La definición y alcance previsto de cada tipo de pruebas se detalla en los subapartados siguientes.

Asimismo, la estrategia contempla la integración de métricas de calidad (como la cobertura de pruebas, el número de defectos abiertos y el tiempo medio de resolución) con el fin de proporcionar una visión objetiva del estado del producto en cada iteración. Los resultados obtenidos retroalimentan el proceso, potenciando la mejora continua tanto en la calidad del desarrollo como en la eficacia del proceso de pruebas.



6.1. Subprocesos de pruebas

Las pruebas que conforman la estrategia integral de calidad en el proyecto THOT pueden clasificarse en varios niveles y tipos, cada uno con objetivos y alcances específicos que, en conjunto, garantizan la fiabilidad y la continuidad del producto. Los siguientes apartados definen cada uno de los niveles de pruebas considerados en el plan de pruebas.

6.1.1. Pruebas unitarias

Las pruebas unitarias se enfocan en verificar el correcto funcionamiento de componentes individuales del código, como funciones, clases o módulos. Su propósito es asegurar que cada parte aislada del sistema realice las operaciones previstas conforme a su diseño. Estas pruebas estarán automatizadas y se ejecutarán de manera continua dentro de la integración continua (CI), permitiendo identificar de inmediato defectos generados por nuevos desarrollos o refactorizaciones. La cobertura amplia de pruebas unitarias contribuirá a mantener la estabilidad del código y a minimizar regresiones en etapas posteriores.

La estrategia de pruebas unitarias en THOT se basará en los siguientes pilares:

- **Herramientas:** uso de *pytest* para Python, *Jest* para Node.js y *go test* para Go, junto con medición de cobertura a través de *Coverage.py*, *Istanbul/nyc* y *go tool cover*.
- **Automatización:** ejecución automática en los pipelines de CI/CD en cada commit, generando reportes de resultados de forma continua.
- **Criterios de aceptación:** se requerirá una cobertura de código igual o superior al 90% y que todas las pruebas unitarias se ejecuten exitosamente antes de permitir la integración.
- **Métricas:** se evaluará la cobertura de sentencias y ramas, el número total de pruebas ejecutadas y los defectos identificados por cada componente.

6.1.2. Pruebas de Integración

Las pruebas de integración validan la comunicación y coherencia entre los diferentes módulos o servicios del sistema. A través de ellas se detectarán errores derivados de interfaces incorrectas, incompatibilidades de datos o dependencias inadecuadas entre componentes. Este tipo de pruebas asegurará que las nuevas funcionalidades se integren correctamente con las existentes. La automatización parcial de estas pruebas es habitual para reducir tiempos y estandarizar la validación de procesos críticos.

La estrategia de pruebas de integración en THOT se estructurará de la siguiente manera:

- **Herramientas:** se utilizará *Pact* para validar contratos entre microservicios, *Postman* para pruebas de APIs REST/gRPC, *integration_test* para aplicaciones Flutter/Dart y *Selenium* para ejecutar flujos integrados en la interfaz de usuario.
- **Automatización:** las pruebas se ejecutarán en entornos *staging* mediante pipelines de CI/CD, generando reportes automáticos de los resultados.
- **Criterios de aceptación:** se exige validar el 100 % de los endpoints críticos, asegurar tiempos de respuesta en pruebas integradas inferiores a 500 ms bajo carga nominal y garantizar la ausencia de defectos bloqueantes antes del despliegue.
- **Métricas:** se medirán la cobertura de pruebas de integración por servicio, la cantidad de contratos verificados sin fallos, así como la latencia y el throughput en escenarios integrados.

6.1.3. Pruebas funcionales

Las pruebas funcionales evaluarán si la aplicación satisface los requisitos establecidos en las historias de usuario y los criterios de aceptación definidos para el sprint. Estas pruebas buscan

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

reproducir el comportamiento esperado desde la óptica del usuario, comprobando flujos de trabajo, validaciones y reglas de negocio, y validando que los desarrollos del sprint cumplen con la funcionalidad comprometida.

La estrategia de pruebas funcionales en THOT se basará en los siguientes puntos:

- **Herramientas:** uso de Selenium para la validación funcional de aplicaciones web y Postman para la verificación funcional de APIs.
- **Automatización:** integración de las pruebas en los pipelines de CI/CD, con generación automática de reportes y registro de la trazabilidad en ClickUp.
- **Criterios de aceptación:** alcanzar el 100% de los criterios establecidos en las historias de usuario, asegurar la validación de los flujos críticos bajo condiciones realistas de operación y garantizar la ausencia de defectos bloqueantes al cierre del sprint.
- **Métricas:** seguimiento del número total de pruebas funcionales ejecutadas, porcentaje de casos exitosos frente a fallidos y tiempo promedio de ejecución por escenario.

6.1.4. Pruebas de aceptación

Las pruebas de aceptación se realizarán en forma de demostraciones funcionales al cliente en hitos concretos del proyecto, con el objetivo de obtener feedback y conformidad sobre los desarrollos entregados, todo ello sobre la plataforma de demostrador. Estas sesiones se apoyarán en los resultados y métricas obtenidas en las pruebas ejecutadas por el equipo durante los sprints (funcionales, no funcionales y de seguridad). La validación se formalizará mediante acta de conformidad y evidencias generadas automáticamente desde las herramientas de CI/CD.

6.1.5. Pruebas de regresión

Las pruebas de regresión verificarán que las nuevas implementaciones no introduzcan defectos en funcionalidades previamente validadas, apoyándose en automatizaciones dentro de los pipelines CI/CD para reutilizar scripts unitarios, funcionales e integrados, reducir tiempos, asegurar consistencia y mantener una cobertura superior al 90% en cada iteración dentro del enfoque Agile.

6.1.6. Pruebas de Ciberseguridad

Las pruebas de ciberseguridad garantizarán que todos los componentes de la plataforma THOT están alineadas con los requisitos de seguridad establecidos para un sistema clasificado bajo ENS categoría Alta y con las buenas prácticas de ISO/IEC 27001 e ISO/IEC 27034 para desarrollo seguro. Estas pruebas incluirán análisis de vulnerabilidades automáticos y recurrentes sobre el código y las dependencias software mediante herramientas como Trivy y OWASP Dependency Check, auditorías estáticas y dinámicas (SAST/DAST), validación de configuraciones seguras en contenedores y servicios de la arquitectura basada en microservicios, así como pruebas de penetración periódicas orientadas a identificar vectores de ataque relevantes (control de acceso, escalado de privilegios, inyección, seguridad de APIs, manipulación de datos sensibles, MITM y exposición de servicios). Adicionalmente, se verificará la correcta aplicación de mecanismos de cifrado en tránsito y en reposo, la robustez

de los sistemas de autenticación y autorización, la integridad del ledger de cadena de custodia y la resiliencia de los servicios frente a ataques de denegación de servicio. En la medida que resulte posible, las pruebas se integrarán en la cadena CI/CD del proyecto, permitiendo ejecutar validaciones automáticas en cada iteración y asegurar la remediación temprana de vulnerabilidades críticas. Los resultados se documentarán en los informes de resultados de cada sprint, así como en informes específicos de ciberseguridad (sobre hitos planificados) y servirán como referencia para las actividades de refuerzo de seguridad y conformidad exigidas en los hitos del proyecto.

6.1.7. Pruebas de Restauración de Datos (Copias de Seguridad)

Las pruebas de restauración de datos verificarán la capacidad del sistema THOT para recuperar información crítica a partir de las copias de seguridad generadas en los distintos entornos, garantizando así el cumplimiento de los requisitos de resiliencia y disponibilidad establecidos para sistemas clasificados bajo ENS categoría Alta. Estas pruebas contemplarán la ejecución periódica de restauraciones controladas —totales y parciales— sobre entornos específicamente habilitados, validando la integridad, consistencia y completitud de los datos restaurados, así como la correcta recuperación de servicios asociados. Se evaluarán además los tiempos de recuperación obtenidos (RTO) y la antigüedad máxima aceptable de los datos restaurados (RPO), contrastándolos con los objetivos definidos para el proyecto. De forma complementaria, se comprobará la validez de los procedimientos de respaldo, la trazabilidad de las copias y la existencia de réplicas seguras almacenadas en repositorios independientes. Los resultados obtenidos se documentarán dentro de los informes de pruebas de ciberseguridad y continuidad, permitiendo identificar desviaciones, establecer medidas correctoras y asegurar la máxima robustez y confiabilidad del sistema ante escenarios de fallo o pérdida de datos.

6.1.8. Pruebas de Continuidad del Negocio

Las pruebas de continuidad del negocio permitirán verificar la capacidad del sistema THOT para mantener operatividad ante escenarios de indisponibilidad parcial o total, alineándose con el cumplimiento de los requisitos establecidos para sistemas clasificados bajo ENS categoría Alta. Estas pruebas evaluarán la eficacia de los procedimientos de contingencia definidos en el plan de continuidad, incluyendo la activación de servicios redundantes, la conmutación automática o manual hacia infraestructuras alternativas, la recuperación operativa en condiciones degradadas y la capacidad del sistema para preservar la integridad, trazabilidad y disponibilidad de los datos durante y después del incidente. Asimismo, se validarán los parámetros de desempeño asociados a los objetivos de continuidad —como el Recovery Time Objective (RTO) y el Recovery Point Objective (RPO)— mediante simulaciones controladas de fallos críticos (caída de microservicios esenciales, pérdida de nodos del clúster, interrupciones de red o fallos en los sistemas de almacenamiento). Las pruebas incluirán también la verificación de los mecanismos de notificación, escalado y registro de eventos, asegurando que la respuesta a incidentes sea coherente con los procedimientos establecidos por la UTE y con las directrices del Plan Director de Seguridad. Los resultados de estas pruebas se integrarán en los informes específicos de ciberseguridad y continuidad, permitiendo

identificar brechas, actualizar procedimientos y garantizar un nivel de resiliencia adecuado para la operación del sistema en entornos críticos.

6.2. Entregables de prueba

Los entregables derivados del plan de pruebas tienen como objetivo documentar de manera sistemática los resultados obtenidos en las actividades de verificación y validación, asegurando la trazabilidad con los requisitos funcionales y técnicos definidos en los documentos de diseño (F1.1.1, F1.1.2, F1.1.3). Estos entregables se generarán de forma iterativa, alineados con la planificación del proyecto y los sprints definidos en la metodología ágil.

Todos los entregables contendrán anexos generados automáticamente desde las herramientas de CI/CD utilizadas en el proyecto. Los informes por sprint se emitirán cada cuatro semanas, mientras que los informes específicos (rendimiento, interoperabilidad, IA, ciberseguridad y continuidad) se elaborarán en los hitos definidos en el cronograma.

Cada entregable incluirá referencias cruzadas a épicas, historias de usuario, componentes involucrados, requisitos funcionales y técnicos, casos de prueba y criterios de aceptación definidos en este plan, asegurando la coherencia y la cobertura completa del alcance previsto.

6.2.1. Informes por sprint

Al cierre de cada sprint se elaborará un informe que refleje el estado de las pruebas realizadas. Este documento incluirá el número total de pruebas ejecutadas, desglosadas por componente y tipo (unitarias, integración, funcionales, regresión), el porcentaje de casos superados y fallidos, la evolución de la cobertura de código y el número y nivel de criticidad de los defectos abiertos y resueltos. Asimismo, se incorporarán comparativas y tendencias respecto al sprint anterior para facilitar la toma de decisiones.

En función del alcance del sprint, estos informes detallarán específicamente los siguientes aspectos:

Interoperabilidad

Se proporcionará información relativa a la validación de la correcta integración entre los distintos componentes del sistema. El informe recogerá los resultados de las pruebas realizadas sobre las interfaces utilizadas en la implementación del sprint, prestando especial atención a escenarios con condiciones de red degradada y a la verificación de mecanismos de sincronización offline. También se documentará el cumplimiento de los contratos de API mediante la herramienta Pact.

Validación de IA

Se generarán evidencias que respalden la validación de los modelos de inteligencia artificial, incluyendo métricas de precisión, robustez y explicabilidad (XAI), así como la verificación del mecanismo de “Sello Triple de Confianza”. Los informes garantizarán la trazabilidad y la integridad de todo el proceso de validación.

6.2.2. Informes de pruebas no funcionales

En hitos específicos se generarán informes que recojan los resultados de pruebas de rendimiento, escalabilidad y resiliencia. Estos informes detallarán métricas clave como

latencia, throughput y consumo de recursos, verificando el cumplimiento de los objetivos de nivel de servicio (SLOs) definidos en el proyecto.

6.2.3. *Informes de pruebas de ciberseguridad y continuidad*

Los informes de pruebas de ciberseguridad y continuidad recogerán de forma integrada los resultados obtenidos en las actividades de análisis de vulnerabilidades, auditorías SAST y DAST, validación de configuraciones seguras, pruebas de penetración, verificación de cifrado, seguridad de APIs y revisión de los controles para sistemas clasificados bajo ENS categoría Alta, así como las evidencias derivadas de las pruebas de restauración de datos —incluyendo la comprobación de integridad, consistencia y la evaluación de los objetivos RTO y RPO— y de las pruebas de continuidad del negocio, que abarcarán simulaciones de fallos críticos, conmutación a infraestructuras alternativas, operación en condiciones degradadas y verificación de los mecanismos de notificación y gestión de incidentes; estos informes se elaborarán en los hitos definidos del proyecto, incorporarán evidencias automáticas provenientes de las herramientas CI/CD y presentarán una valoración global de la seguridad y resiliencia del diseño del sistema, el seguimiento de vulnerabilidades y riesgos residuales, y las recomendaciones necesarias para garantizar la robustez y conformidad del sistema THOT.

6.3. Técnicas de diseño de pruebas

Para garantizar la calidad del software y una cobertura adecuada, se emplearán diversas técnicas de diseño de pruebas. La elección de cada técnica dependerá del tipo de componente, la funcionalidad bajo prueba y los objetivos específicos de cada fase. A continuación, se describen las técnicas contempladas en el proyecto.

En primer lugar, se utilizarán técnicas de caja negra, centradas en validar el comportamiento del sistema sin considerar su estructura interna. Estas técnicas son la base del diseño de pruebas funcionales. Entre ellas, la partición de equivalencia permitirá dividir los datos de entrada en clases que se espera produzcan resultados similares, reduciendo así el número de casos de prueba sin comprometer la cobertura. Complementariamente, se aplicará el análisis de valores límite, que se enfoca en verificar los valores situados en los extremos de las particiones de equivalencia, ya que es en estos puntos donde suelen aparecer errores. Para componentes con múltiples reglas o condiciones, se recurrirá a la tabla de decisiones, ya que facilita la representación de combinaciones de entradas y sus acciones resultantes. Finalmente, en componentes que gestionen flujos definidos, se empleará la técnica de transición de estados, que permite comprobar cómo el software cambia entre diferentes estados y asegura que las transiciones se realicen correctamente.

Por otro lado, se aplicarán técnicas de caja blanca, orientadas a analizar la estructura interna del código. Estas técnicas garantizan que el código se ejecute de manera completa y correcta. Se utilizará la cobertura de sentencias, que asegura que cada línea de código ejecutable se ejecute al menos una vez mediante herramientas como Coverage.py (Python), Istanbul/nyc (JavaScript/Node.js) y go cover (Go), y la cobertura de decisiones, que verifica que todas las ramas de las estructuras de control, tanto en sus condiciones verdaderas como falsas, sean evaluadas con las mismas herramientas configuradas para branch coverage.

Además, se incorporarán técnicas híbridas y basadas en experiencia, que combinan enfoques funcionales y estructurales. La principal será la prueba basada en casos de uso, que diseña los

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

casos de prueba a partir de escenarios reales que un usuario seguiría para interactuar con el sistema. Este procedimiento será prioritario, ya que permite vincular las pruebas directamente con los requerimientos funcionales. También se realizarán pruebas exploratorias, en las que el tester explora el software mientras lo prueba, diseñando casos sobre la marcha para descubrir comportamientos inesperados. Para optimizar la cobertura en combinaciones de parámetros, se aplicará la técnica de pruebas por pares (pairwise testing), que valida todas las combinaciones posibles de dos parámetros. Para ello, se utilizará la herramienta Microsoft PICT.

Finalmente, se consideran otras técnicas relevantes que complementarán el proceso. Las pruebas de regresión se ejecutarán para asegurar que los cambios en el software no afecten funcionalidades existentes. Las pruebas basadas en riesgos permitirán priorizar la verificación de áreas críticas o vulnerables del sistema. Asimismo, se realizarán pruebas estáticas, que incluyen la revisión de código y documentos sin ejecutar el programa, mediante inspecciones y análisis automatizados. Para este propósito, se utilizará la herramienta SonarQube.

6.4. Métricas a recopilar

Con el fin de asegurar la calidad, la seguridad y el rendimiento del sistema HORUS, así como optimizar el proceso de pruebas, se recopilarán métricas que permitan evaluar de manera objetiva el nivel de cumplimiento de los criterios de aceptación y de los objetivos establecidos en el plan. Todas las métricas se integrarán en los pipelines de CI/CD para su recogida automática y se incorporarán tanto en los informes de cada sprint como en los entregables previstos en el cronograma del proyecto. Estos indicadores serán fundamentales para apoyar las decisiones Go/No-Go en cada iteración y garantizarán la trazabilidad respecto a los requisitos funcionales y técnicos definidos.

Categoría	Métrica	Descripción	Umbral	Herramienta recomendada	Tipo de prueba
Cobertura	Cobertura de código	% de sentencias y ramas ejecutadas	>=90%	coverage.py / Istanbul (nyc) / go tool cover	Unitarias / Regresión
Cobertura	Cobertura funcional	Requisitos vs pruebas ejecutadas	100%	ClickUp	Funcionales / Aceptación
Cobertura	Cobertura de endpoints críticos	Endpoints validados en integración	100%	Postman / Pact / CI/CD	Integración
Cobertura de seguridad	Cobertura SAST/DAST	Validación de endpoints críticos por	>=90%	SonarQube / OWASP ZAP	Seguridad

		análisis estático/dinámico			
Ejecución	Casos ejecutados	Número total por tipo	N/A	pytest / Jest / go test / Selenium / Appium	Todas
Ejecución	% casos superados	Casos exitosos frente a fallidos	>95%	ClickUp / CI/CD	Todas
Ejecución	Tiempo medio por escenario	Duración media por caso funcional	<2 min	Selenium / Appium	Funcionales / Aceptación
Ejecución	Casos reejecutados	Número de pruebas relanzadas	Tendencia decreciente	ClickUp	Todas
Calidad	Defectos por severidad	Defectos abiertos/cerrados por criticidad	0 críticos	ClickUp	Todas
Calidad	TMR de defectos	Tiempo medio de resolución	<48 h	ClickUp	Proceso
Calidad	Tendencia de defectos	Evolución de defectos respecto a sprints	Decreciente	ClickUp / CI/CD	Proceso
Seguridad	Vulnerabilidades detectadas	Nº por severidad (crítica/alta/media)	0 críticas	SonarQube / Trivy / OWASP DC / ZAP	Seguridad
Seguridad	Respuesta a ataque	Tiempo máximo ante MITM / inyección	<2 s	Wireshark / Burp Suite / OWASP ZAP	Seguridad
Seguridad	Validación criptográfica	Cifrado en tránsito y reposo	Conforme	OpenSSL / Scripts CI/CD	Seguridad
Seguridad	Transacciones en ledger	Registros generados en libro inmutable	100% registradas	Ledger THOT / hashing	Seguridad / Custodia
Cumplimiento	Evidencias ENS	Cifrado, trazabilidad y	Conforme	Scripts CI/CD / Auditorías	Seguridad

		controles ENS Alto			
Rendimiento	Latencia API	Media y p95/p99 en endpoints críticos	<500 ms	k6	Rendimiento / Carga
Rendimiento	Throughput	Transacciones por segundo bajo carga	Según SLO	k6	Carga / Estrés
Rendimiento	Consumo de recursos	CPU, RAM, IO bajo carga	Según SLO	k6 / Prometheus / Grafana	Rendimiento
Interoperabilidad	Robustez en red degradada	Reconexión y reconciliación de datos	100% reconexión	Scripts / K6	Integración
IA	Precisión del modelo	% de acierto en escenarios operativos	>=90%	Scripts de evaluación	IA
IA	FPR/FNR	Falsos positivos y negativos	<5%	Scripts de evaluación	IA
IA	Tiempo de inferencia	Tiempo medio por predicción	<2 s	Logs Edge / Scripts	IA / Rendimiento
IA	Sello Triple de Confianza	Hash input/model/output	100% verificado	Scripts / Hashing	IA / Seguridad
IA	Robustez adversarial	Precisión tras ataques adversariales	>=90%	Adversarial Robustness Toolbox	IA Seguridad
Proceso	Automatización de pruebas	% de pruebas automatizadas	>=80%	pytest / Jest / selenium / appium / integration_test	Unitarias / Funcionales / Regresión
Proceso	Salud CI/CD	Ejecuciones exitosas	>95%	Pipelines CI/CD	Proceso
Proceso	Deuda técnica	Defectos acumulados vs resueltos	Decreciente	SonarQube / ClickUp	Proceso

Interoperabilidad	Validación de contratos	Contratos de API validados	100%	Pact	Integración
Interoperabilidad	Sincronización offline	Reconciliación tras reconexión	100%	Scripts / mecanismos offline	Integración

6.5. Requisitos de datos de prueba

Los datos de prueba se utilizarán para validar el alcance definido en cada uno de los sprints, por lo que su naturaleza y composición estarán directamente vinculadas al contexto funcional y técnico de cada iteración. De este modo, la generación, uso y mantenimiento de los datos se alineará con los objetivos específicos de validación establecidos en cada ciclo de desarrollo. Siempre que resulte posible, se solicitará al cliente la provisión de ejemplos de datos reales o representativos que faciliten la adecuada verificación de las funcionalidades desarrolladas. Dichos datos deberán ser previamente anonimizados conforme a las políticas de protección de datos aplicables, garantizando la ausencia de información sensible o personal identificable.

A partir del material proporcionado, y cuando sea necesario ampliar la cobertura o reproducir escenarios complejos, se emplearán herramientas semiautomáticas para la generación de datasets sintéticos adaptados a los requisitos del sistema bajo prueba. Estas herramientas permitirán preservar la coherencia interna de los datos, controlar su variabilidad y reutilizarlos de forma consistente en los distintos entornos de prueba. En este contexto, se utilizarán soluciones como *Synthetic Data Vault (SDV)*, que facilita la generación de datos tabulares y relacionales mediante modelos probabilísticos y generativos, o *Faker*, orientada a la creación rápida de datos ficticios que cubran perfiles de usuarios, fechas, registros administrativos y otros atributos necesarios para pruebas funcionales y de validación.

Cuando las pruebas requieran validar capacidades relacionadas con biometría o análisis de imágenes, y siempre dentro de los márgenes permitidos por la normativa aplicable, el proyecto podrá apoyarse en datasets públicos ampliamente utilizados en investigación. Entre ellos se encuentran el repositorio abierto de huellas dactilares disponible en GitHub, que proporciona múltiples colecciones útiles para ensayos de calidad, normalización y cotejo, así como los *NIST Biometric Special Databases and Software*, que incluyen conjuntos de datos de huellas dactilares, iris, rostro y manuscritos empleados como referencia internacional para la evaluación de algoritmos biométricos y verificación de estándares.

En caso de no disponer de datos reales de referencia, los conjuntos de prueba se generarán íntegramente de forma sintética tomando como base las fuentes de información disponibles en el proyecto(por ejemplo, los materiales de jornadas de formación). Este enfoque permitirá representar escenarios operativos verosímiles y cubrir los flujos funcionales más significativos sin comprometer la protección de datos.

Todos los datasets generados, tanto reales anonimizados como sintéticos o procedentes de fuentes públicas, serán almacenados bajo control de versiones y quedarán vinculados formalmente a los distintos entornos de prueba definidos en el proyecto. Tanto los datos como las configuraciones de los entornos podrán evolucionar conforme avance el desarrollo y se amplíe el alcance funcional previsto para los sprints siguientes, asegurando la trazabilidad, la reproducibilidad y la consistencia de las validaciones realizadas.

6.6. Requisitos del entorno de pruebas

El proceso de verificación y validación del sistema se apoyará en tres entornos diferenciados, diseñados para cubrir las distintas fases del ciclo de pruebas. Cada entorno permitirá reproducir con fidelidad las condiciones necesarias para ejecutar los distintos tipos de prueba previstos (unitarias, de integración, de rendimiento y de aceptación).

Los entornos definidos son los siguientes:

- Entorno de Desarrollo Local (On-premise): alojado en las instalaciones de cada empresa participante.
- Entorno de Preproducción (Staging): desplegado en infraestructura común a todos los socios de la UTE.
- Entorno de Demostrador Cliente (Producción Controlada): entorno estable que emulará el entorno productivo y estará destinado a pruebas de aceptación y demostración funcional ante Policía Científica.

El entorno de desarrollo local estará ubicado en las instalaciones de cada una de las entidades que participan en el proyecto y servirá como base para la validación inicial de los componentes desarrollados. Cada participante dispondrá de su propio entorno, configurado de acuerdo con las directrices técnicas comunes y con los parámetros establecidos por la UTE para asegurar la compatibilidad entre las distintas implementaciones. Las configuraciones deberán conservarse bajo control de versiones, garantizando la trazabilidad de los artefactos ejecutables y de los parámetros de despliegue utilizados. Las versiones de las aplicaciones y dependencias se identificarán mediante un sistema unificado de etiquetado y registro, de forma que sea posible reconstruir en cualquier momento las condiciones bajo las cuales se obtuvo un determinado resultado de prueba. El acceso a este entorno quedará restringido al personal de desarrollo y verificación designado por cada entidad, conforme a las políticas internas de seguridad y confidencialidad que apliquen a nivel organizativo. Aunque este entorno se gestionará de manera independiente por cada socio, deberá mantener la coherencia con la arquitectura de referencia del proyecto, empleando los mismos repositorios de código fuente, versiones de librerías homologadas y mecanismos comunes de documentación y seguimiento.

El entorno de preproducción (o staging) se desplegará sobre la infraestructura común gestionada por la UTE, basada en un cluster Kubernetes operativo sobre OpenStack. Este entorno constituirá el punto de integración técnica entre los distintos módulos y servicios desarrollados por las entidades participantes, y servirá para realizar ensayos en condiciones funcionales y de carga cercanas a las de un sistema operativo. Su configuración estará definida mediante ficheros de despliegue declarativos en Kubernetes, que determinarán de forma precisa las versiones de las imágenes, las variables de configuración, los recursos asignados y las dependencias entre servicios. Dichos ficheros constituirán la referencia oficial del entorno y se mantendrán bajo control de versiones, asegurando la trazabilidad de toda modificación y posibilitando la reproducción de un estado exacto del sistema en cualquier momento. Los despliegues se ejecutarán a través de procedimientos automatizados de integración y entrega continua (CI/CD).

El entorno de producción, concebido como entorno demostrador para el cliente, se desplegará en un segundo clúster Kubernetes sobre el entorno OpenStack gestionado por la UTE. El clúster Kubernetes será independiente del utilizado por el entorno de preproducción. Esta independencia garantizará la estabilidad operativa y el aislamiento entre los entornos, evitando interferencias durante la ejecución de pruebas o la validación funcional. Cada entorno dispondrá de sus propios recursos de red, almacenamiento, cuentas de servicio y configuraciones, de modo que las operaciones en uno no afecten al otro. La definición del entorno de producción se realizará mediante los mismos mecanismos declarativos que en preproducción, manteniendo así la coherencia técnica del proyecto, pero gestionando los repositorios, registros de imágenes y parámetros de despliegue de forma separada. Este entorno albergará únicamente versiones estables del software, previamente verificadas y validadas, y su propósito será permitir al cliente interactuar con el sistema en condiciones controladas para la ejecución de pruebas de aceptación y demostraciones funcionales. La operación, mantenimiento y control de acceso estarán a cargo de la UTE, que garantizará la integridad y trazabilidad de las versiones desplegadas y la disponibilidad del sistema durante el periodo de evaluación.

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

7. Actividades de Prueba y Estimaciones

Las pruebas principales ejecutadas durante los sprints se centrarán en la validación funcional de los desarrollos incorporados en cada iteración, buscando asegurar que los entregables cumplan con los criterios de calidad y los objetivos definidos en el alcance planificado.

De forma complementaria, se llevarán a cabo pruebas periódicas de carácter no funcional —incluyendo ensayos de escalabilidad, estrés y carga— en el entorno de staging, con el propósito de verificar la solidez estructural y el desempeño del sistema de manera independiente a las funcionalidades implementadas recientemente. Estas actividades resultan necesarias para evaluar condiciones de alto consumo de recursos y escenarios de uso intensivo que no pueden abordarse dentro del ciclo regular de los sprints.

Durante los periodos en que se programen dichas pruebas no funcionales, la planificación de los sprints contemplará un alcance funcional reducido, a fin de liberar capacidad del equipo y garantizar la correcta ejecución y análisis de los resultados obtenidos.

8. Dotación de Personal

La dotación de personal es un factor crítico para garantizar la calidad del proyecto y la correcta ejecución del plan de pruebas. En un sistema complejo como THOT, basado en microservicios, inteligencia artificial, almacenamiento inmutable y comunicaciones avanzadas, la calidad no depende únicamente del diseño del plan, sino de contar con equipos adecuados en número, competencias y coordinación. Una dotación equilibrada permite cubrir todos los niveles de prueba (unitarias, integración, sistema, aceptación, interoperabilidad, rendimiento y ciberseguridad) con la profundidad y el ritmo que exige la metodología ágil, reduciendo riesgos técnicos y costes derivados de defectos tardíos.

La asignación correcta de roles incide directamente en:

- **Especialización técnica**: validar IA responsable, trazabilidad y pruebas de interoperabilidad en escenarios críticos.
- **Cumplimiento normativo y seguridad**: integrar controles ENS Alto, ISO 27001 y EU AI Act desde el diseño de las pruebas.

Por ello, esta sección define los roles, responsabilidades, nivel de implicación y necesidades de capacitación para garantizar que el proceso de pruebas alcance los estándares de calidad esperados.

8.1. Roles, actividades y responsabilidades

El equipo de pruebas estará compuesto por perfiles multidisciplinares distribuidos entre las entidades de la UTE.

Rol	Organizaciones	Responsabilidades Principales	Implicación
Responsable de Pruebas	HI IBERIA	Planificar, coordinar y supervisar todas las actividades de prueba Validar la estrategia y los entregables Gestionar riesgos asociados al proceso de pruebas	Alta, durante todo el ciclo de vida
Equipo de Testing	HI IBERIA, ETRA, HERTA, LabLENI, IMBOX	Diseñar, ejecutar y documentar casos de prueba Automatizar pruebas unitarias e integración Realizar pruebas funcionales y no funcionales Reportar incidencias y colaborar en su resolución	Alta, en Fase 2 y Fase 3

Especialistas en Ciberseguridad	HARDLINK	Ejecutar pruebas de seguridad (pentesting, auditorías técnicas) Validar cumplimiento normativo (ENS Alto, ISO 27001) Supervisar pruebas relacionadas con protección de datos y trazabilidad	Media, en hitos específicos
Desarrolladores	HI IBERIA, ETRA, HERTA, LabLENI, IMBOX	Atender incidencias detectadas en pruebas Corregir defectos y colaborar en la preparación de entornos	Media, durante todo el ciclo
Gestor de Proyecto	ETRA	Asegurar recursos y aprobar el plan de pruebas Gestionar riesgos y dependencias	Alta, en planificación y seguimiento
Cliente	Policía Científica, CDTI	Validar resultados de pruebas de aceptación Participar en pruebas funcionales críticas	Implicación en hitos de validación

8.2. Necesidades de capacitación

El equipo recibirá formación en:

- Metodología Agile aplicada a QA.
- Herramientas de automatización y monitorización (Selenium, JMeter, K6, Pact, SonarQube, Trivy).
- Validación de IA (explicabilidad, sesgos, métricas XAI).
- Verificación de trazabilidad en bases de datos ledger.
- Cumplimiento normativo (ENS Alto, ISO 27001, EU AI Act).

Cofinanciado por la Unión Europea
MINISTERIO DE HACIENDA
Fondos Europeos

9. CRONOGRAMA

El proyecto se desarrollará bajo un enfoque ágil, estructurado en sprints de una duración fija de cuatro semanas cada uno. Este esquema permitirá una planificación iterativa e incremental del producto, garantizando la entrega periódica de versiones funcionales y verificables, así como una retroalimentación continua entre los equipos de desarrollo y pruebas, y una comunicación periódica con Policía Científica para la validación de los resultados parciales.

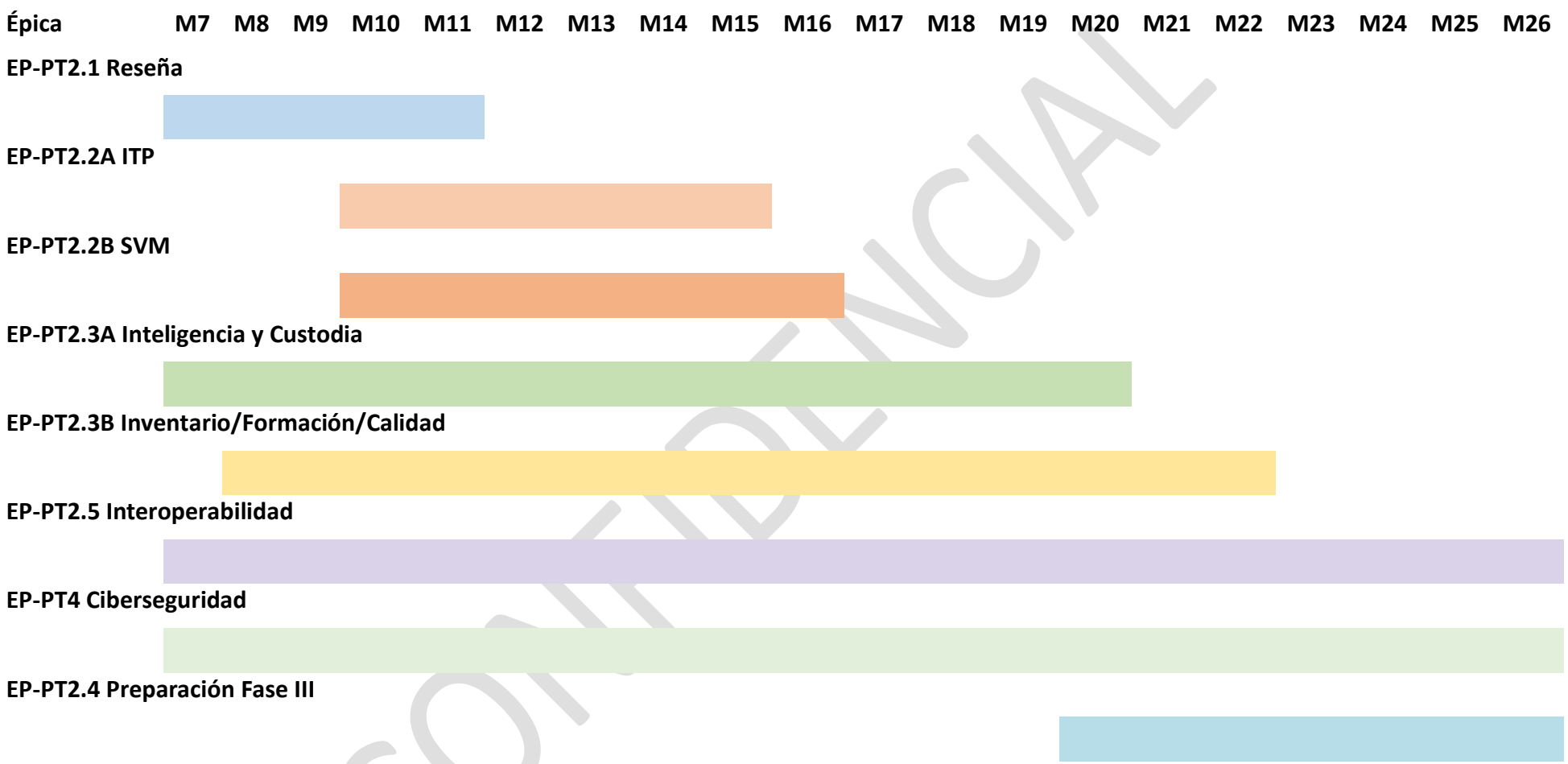
La planificación de los sprints se basará en la definición de épicas, entendidas como conjuntos de funcionalidades de alto nivel que agrupan requerimientos relacionados. Estas épicas se derivan directamente de los paquetes de trabajo propuestos en la oferta, lo que asegura la trazabilidad entre los requerimientos iniciales del proyecto y los desarrollos.

Cada épica se descompondrá en historias de usuario, entendidas como unidades de trabajo funcional de menor alcance que describen una necesidad específica desde la perspectiva del usuario final. Las historias de usuario incluirán, como parte del proceso de pruebas, criterios de aceptación que permitirán verificar el cumplimiento de los requerimientos funcionales y técnicos asociados a cada iteración.

La distribución de las historias de usuario se realizará de forma progresiva a lo largo de los distintos sprints, teniendo en cuenta su prioridad, complejidad, dependencias identificadas y el grado de desarrollo alcanzado por el proyecto en cada iteración. En cada sprint se seleccionarán las historias que puedan ser completadas y probadas dentro del periodo de cuatro semanas, garantizando así que al cierre de cada ciclo se disponga de un incremento del producto validado, documentado y potencialmente desplegable.

El presente plan de pruebas incorpora un cronograma preliminar que servirá como base para la planificación del trabajo en las fases posteriores del proyecto. Dicho cronograma se ajustará conforme al avance del proyecto y a los resultados de las revisiones de sprint. De esta manera, se mantendrá la flexibilidad necesaria para incorporar mejoras o ajustes derivados de la evaluación continua del proceso de desarrollo y de la calidad de las entregas intermedias.

El detalle de las épicas e historias de usuario se encuentra detallado en la sección 3 del presente documento.



- Interoperabilidad: validaciones de contrato (M7, M9, M11, M13, M15, M17, M19, M21, M23, M25) y pruebas de red degradada (M8, M10, M12, M14, M16, M18, M20, M22, M24, M26).

- Ciberseguridad: Pentesting (M7, M10, M13, M16, M19, M22, M25) y remediación (M8, M11, M14, M17, M20, M23, M26).

10. TECNOLOGÍAS DE SOPORTE A LAS PRUEBAS

Para garantizar la calidad, seguridad y rendimiento del sistema THOT, se empleará un conjunto de herramientas especializadas que darán soporte a los distintos tipos de prueba definidos en este plan.

En primer lugar, para la automatización de pruebas de interfaz de usuario web se utilizará **Selenium**. Esta herramienta permitirá ejecutar flujos de usuario completos de manera repetible, asegurando la verificación continua del comportamiento funcional en cada iteración del desarrollo.

En lo relativo al análisis estático del código, se integrará **SonarQube** como plataforma centralizada para la detección temprana de errores, vulnerabilidades y malas prácticas, proporcionando métricas de calidad, seguimiento de deuda técnica y verificaciones automáticas sobre cada commit dentro de los pipelines de CI/CD. Su incorporación permitirá mantener un estándar elevado de calidad desde las primeras fases de implementación.

Para las pruebas de seguridad dinámicas, se empleará **OWASP ZAP** como herramienta principal de análisis en tiempo de ejecución. Su utilización permitirá identificar vulnerabilidades tales como inyecciones SQL, XSS y fallos de autenticación o autorización durante la ejecución de los servicios desplegados en los entornos de staging. Complementariamente, la validación de la seguridad en contenedores, imágenes Docker y dependencias se reforzará mediante **Trivy**, que garantizará que los artefactos promovidos hacia entornos superiores cumplan con los niveles de seguridad exigidos para la plataforma.

En el ámbito de las pruebas de rendimiento, carga y estrés, el proyecto empleará **k6**, que permitirá modelar escenarios realistas de concurrencia, medir la latencia y throughput de los servicios y evaluar el consumo de recursos bajo condiciones intensivas. Sus scripts en JavaScript facilitarán la integración en CI/CD y permitirán la ejecución automatizada de pruebas no funcionales en los hitos definidos del cronograma.

La validación de la interoperabilidad entre microservicios se llevará a cabo mediante **PACT**, que permitirá asegurar la estabilidad de los contratos API entre consumidores y proveedores y evitar fallos en la integración derivados de cambios no controlados. Para la verificación funcional de APIs y flujos complejos se utilizará **Postman**, tanto de forma manual como integrada dentro de los pipelines de pruebas.

La gestión de casos de prueba, resultados, evidencias y defectos se organizará mediante **ClickUp**, que actuará como repositorio central de la trazabilidad entre historias de usuario, criterios de aceptación y ejecuciones de prueba. Esta herramienta permitirá mantener la coherencia documental del proceso de QA, así como facilitar el análisis de métricas de avance y calidad en cada sprint.

Finalmente, las pruebas unitarias y la medición de cobertura de código se realizarán mediante las herramientas específicas para cada lenguaje empleado en la solución: **pytest** y **coverage.py** en Python, **Jest** e **Istanbul (nyc)** en Node.js y los mecanismos nativos **go test** y **go tool cover** en Go. Todas estas herramientas se integrarán plenamente en los pipelines CI/CD para asegurar que cada componente cumpla los niveles de cobertura establecidos y que cualquier regresión sea detectada de manera temprana.