GESTIÓN DE PRUEBAS FUNCIONALES, TÉCNICAS Y DE SEGURIDAD

DIRECCIÓN DE GESTIÓN DEL ORDENAMIENTO SOCIAL DE LA PROPIEDAD
SUBDIRECCIÓN DE SISTEMAS DE INFORMACIÓN DE TIERRAS

Marzo, 2025

CONTENIDO

1. INTRODUCCIÓN

La calidad, estabilidad y seguridad de los sistemas de información implementados en la Agencia Nacional de Tierras (ANT) son elementos clave para garantizar la continuidad operativa y la confianza en los servicios prestados a la ciudadanía. En este contexto, el presente instructivo establece las directrices para la planificación, ejecución, documentación y seguimiento de las pruebas funcionales, técnicas y de seguridad aplicables a los productos de software desarrollados, mantenidos o adquiridos por la entidad. Este documento aplica tanto a sistemas de información tradicionales como a sistemas de información geográfica (SIG), y busca consolidar un enfoque integral de aseguramiento de calidad, que permita validar el cumplimiento de los requerimientos definidos, mitigar riesgos técnicos y garantizar la protección de la información institucional.

El instructivo es complementario al documento “Ciclo de Vida de los Sistemas de Información” y debe aplicarse en conjunto con los lineamientos y guías técnicas vigentes, como parte del Sistema Integrado de Gestión de la ANT.

2. OBJETIVOS

Establecer los lineamientos técnicos y metodológicos que deben seguirse en la ejecución de pruebas funcionales, técnicas y de seguridad sobre los sistemas de información desarrollados, adaptados o adquiridos por la Agencia Nacional de Tierras (ANT), con el fin de:

- Validar que las soluciones tecnológicas cumplan con los requerimientos funcionales y no funcionales previamente definidos.
- Verificar el correcto desempeño, interoperabilidad y estabilidad de los sistemas en distintos entornos tecnológicos.
- Garantizar la implementación de controles de seguridad desde las etapas tempranas del desarrollo y antes de su paso a producción, reduciendo riesgos asociados a vulnerabilidades o brechas de seguridad.

Este instructivo contribuye a fortalecer el aseguramiento de la calidad del software institucional y a cumplir con los estándares definidos por la ANT en materia de desarrollo seguro, eficiente y alineado con los objetivos misionales de la entidad.

3. ALCANCE

Este instructivo aplica a todos los sistemas de información desarrollados, adaptados o adquiridos por la Agencia Nacional de Tierras (ANT), tanto tradicionales (alfanuméricos) como geográficos (SIG). Es de uso obligatorio para los equipos técnicos, contratistas y terceros que intervienen en el proceso de verificación de calidad de soluciones tecnológicas en la entidad.
El instructivo cubre:

- Pruebas funcionales para validar el cumplimiento de requerimientos operativos.
- Pruebas técnicas para verificar aspectos de rendimiento, interoperabilidad, integridad de datos y compatibilidad con infraestructura.
- Pruebas de seguridad para identificar y mitigar vulnerabilidades y garantizar la protección de la información institucional.

Aplica desde las fases tempranas del desarrollo hasta el proceso previo al paso a producción, incluyendo despliegues en ambientes de prueba, homologación y validación por parte de usuarios clave.

4. PRINCIPIOS RECTORES

Este instructivo se fundamenta en principios que orientan la adecuada planificación, ejecución y seguimiento de las pruebas sobre los sistemas de información de la Agencia Nacional de Tierras (ANT). Estos principios son:

- **Orientación a la calidad**: Las pruebas deben enfocarse en verificar el cumplimiento de los requisitos funcionales, no funcionales y técnicos establecidos, garantizando que el sistema responda a las necesidades institucionales con altos estándares de calidad.

- **Seguridad integrada**: La seguridad debe ser validada de manera transversal desde las primeras fases de desarrollo, aplicando el enfoque *Security by Design*, asegurando que la confidencialidad, integridad y disponibilidad de la información estén protegidas ante amenazas.

- **Trazabilidad de resultados**: Cada prueba debe estar claramente asociada a requerimientos previamente definidos, permitiendo evidenciar su validación y facilitar la auditoría del proceso.

- **Interoperabilidad y compatibilidad**: Las pruebas deben contemplar la interacción del sistema con otras plataformas internas y externas, asegurando que los servicios tecnológicos se integren sin conflictos ni pérdidas de funcionalidad.

- **Estándares y buenas prácticas**: El proceso de pruebas debe alinearse con metodologías reconocidas, como DevOps y pruebas automatizadas, así como con los lineamientos técnicos definidos en el marco del Sistema Integrado de Gestión (SIG) y del Modelo de Referencia de Arquitectura Empresarial (MRAE).

- **Participación multidisciplinaria**: Las pruebas deben involucrar activamente a analistas, desarrolladores, usuarios clave, especialistas en infraestructura y seguridad, garantizando una visión integral de validación

- **Ciclo de mejora continua**: Los hallazgos y resultados obtenidos deben retroalimentar las fases de diseño y desarrollo, contribuyendo a la madurez de los procesos de construcción de soluciones tecnológicas en la ANT.

5. GLOSARIO DE TÉRMINOS

Ambiente de Pruebas: Entorno controlado en el que se ejecutan pruebas técnicas, funcionales o de seguridad, aislado del ambiente productivo.

Azure DevOps: Plataforma de Microsoft para la gestión del ciclo de vida del desarrollo de software, que incluye funcionalidades como repositorios, integración y entrega continua (CI/CD), gestión de pruebas y seguimiento de requerimientos.

Azure Test Plans: Herramienta de Azure DevOps que permite planificar, ejecutar y rastrear pruebas manuales o exploratorias.

Caso de Prueba: Conjunto de condiciones o acciones diseñadas para validar una funcionalidad específica del sistema, con datos de entrada y resultados esperados.

Ciclo de Vida del Software: Conjunto de fases que abarca el desarrollo, mantenimiento y retiro de un sistema de información, incluyendo análisis, diseño, construcción, pruebas, implementación y cierre.

DevOps: Práctica que integra el desarrollo (Dev) y las operaciones (Ops) para acelerar la entrega de software de calidad mediante automatización, colaboración y entrega continua.

DevSecOps: Extensión de DevOps que incorpora prácticas de seguridad (Sec) en todo el ciclo de desarrollo, garantizando que la seguridad esté integrada desde las fases iniciales del proyecto.

Hallazgo: Resultado de una prueba que evidencia una falla, error o desviación respecto a los criterios de aceptación establecidos.

Herramienta de Pruebas: Software utilizado para diseñar, ejecutar, automatizar o registrar pruebas.

Informe de Resultados de Pruebas: Documento consolidado que presenta los resultados obtenidos en las pruebas ejecutadas, incluyendo los casos aprobados, fallidos, hallazgos y observaciones.

OWASP: Open Web Application Security Project. Comunidad internacional que promueve la seguridad en aplicaciones web, y publica estándares como el OWASP Top 10.

Plan de Pruebas: Documento que define el enfoque, alcance, recursos, cronograma y criterios de aceptación para la ejecución de pruebas en un proyecto.

Prueba Automatizada: Ejecución de pruebas mediante scripts o herramientas sin intervención manual, útil para validar regresiones o ejecutar pruebas repetitivas.

Pruebas Funcionales: Validaciones que aseguran que el sistema cumple con los requerimientos funcionales desde la perspectiva del usuario.

Pruebas Técnicas: Evaluaciones del rendimiento, escalabilidad, compatibilidad, interoperabilidad e integridad técnica del sistema.

Pruebas de Seguridad: Validaciones orientadas a detectar vulnerabilidades, debilidades o brechas que puedan comprometer la confidencialidad, integridad o disponibilidad de la información.

Re-ejecución: Proceso de volver a ejecutar los casos de prueba afectados por un hallazgo, con el fin de validar si la corrección fue exitosa.

Reporte de Hallazgos: Registro formal de errores, fallas o desviaciones detectadas durante las pruebas, que incluye severidad, responsable y estado.

Sistema de Información Geográfica (SIG): Tipo de sistema que permite capturar, almacenar, analizar y visualizar datos geoespaciales.

UAT (Aser Acceptance Testing): Pruebas de aceptación de usuario realizadas por el área funcional o usuaria para validar que la solución cumple con sus requerimientos antes de ser liberada.

6. METODOLOGÍA PARA LA GESTIÓN DE PRUEBAS

6.1 Tipo de Pruebas

La selección de los tipos de pruebas a ejecutar estará determinada por la metodología de desarrollo adoptada en el proyecto (ágil, tradicional o híbrida), los criterios de aceptación establecidos, así como por la disponibilidad de capacidades técnicas y herramientas de automatización en la entidad. A continuación, se describen los tipos de pruebas más relevantes:

- **Pruebas unitarias**: Validan el funcionamiento correcto de componentes individuales del sistema (por ejemplo, funciones, métodos o clases) de manera aislada, asegurando que cada unidad cumpla su propósito específico.
- **Pruebas de integración**: Evalúan la correcta interacción entre diferentes módulos, servicios o capas del sistema, como la comunicación entre el frontend y el backend, o entre la base de datos y las interfaces API.
- **Pruebas funcionales**: Permiten verificar que el sistema responde adecuadamente a los requerimientos funcionales definidos, enfocándose en el comportamiento observable desde la perspectiva del usuario final.
- **Pruebas de rendimiento**: Analizan la capacidad del sistema para operar eficientemente bajo diferentes condiciones de carga, midiendo tiempos de respuesta, consumo de recursos, estabilidad y escalabilidad.
- **Pruebas de seguridad**: Tienen como finalidad identificar vulnerabilidades potenciales, asegurando el cumplimiento de políticas de autenticación, autorización, cifrado y protección de la información sensible.
- **Pruebas de usabilidad**: Evalúan la experiencia del usuario, valorando aspectos como la facilidad de navegación, la claridad en el diseño de las interfaces y la accesibilidad.
- **Pruebas de regresión**: Verifican que las nuevas funcionalidades incorporadas o ajustes realizados no hayan generado efectos negativos o alteraciones en las funcionalidades ya existentes.
- **Pruebas de carga**: Simulan múltiples usuarios o transacciones simultáneas para analizar la estabilidad y el rendimiento del sistema bajo condiciones de alta demanda.
- **Pruebas de aceptación de usuario (UAT)**: Son ejecutadas por los usuarios funcionales con el fin de validar que el sistema cumple sus expectativas, necesidades y requerimientos antes de ser liberado al ambiente de producción.

6.2 Fases de la Gestión de Pruebas

El proceso de gestión de pruebas en la Agencia Nacional de Tierras se estructura en fases que permiten planificar, ejecutar y controlar de manera efectiva las validaciones técnicas, funcionales y de seguridad de los sistemas de información, sean estos alfanuméricos o geográficos. Las fases descritas a continuación son adaptables al modelo de desarrollo adoptado en cada proyecto.

1. **Planeación de pruebas:** Se definen los objetivos de la prueba, su alcance, los tipos de prueba a ejecutar, los criterios de aceptación, roles responsables y los entornos requeridos. Esta fase contempla también la definición de herramientas y cronograma de ejecución.

2. **Diseño de casos y escenarios de prueba:** A partir de los requerimientos funcionales y no funcionales aprobados, se diseñan los casos de prueba, describiendo entradas, condiciones, pasos a seguir, datos requeridos y resultados esperados. Se deben incluir pruebas positivas y negativas.

3. **Preparación del entorno y datos de prueba:** Se configura el ambiente de pruebas (infraestructura, servicios, bases de datos) y se generan los datos requeridos, ya sean simulados o anonimizado de datos reales, garantizando la no exposición de información sensible.

4. **Ejecución de pruebas:** Se ejecutan los casos diseñados, registrando los resultados reales frente a los esperados. Esta fase puede involucrar pruebas manuales y/o automatizadas, según la naturaleza del sistema y los recursos disponibles.

5. **Registro y gestión de hallazgos:** Los errores, fallas o desviaciones detectadas se documentan en herramientas de seguimiento como Azure DevOps o similares. Se clasifican por nivel de criticidad, se prioriza su atención y se hace trazabilidad del ciclo de corrección.

6. **Re ejecución y validación:** Una vez corregidas las incidencias reportadas, se repiten los casos afectados para verificar que los ajustes hayan sido efectivos y no hayan generado efectos colaterales (pruebas de regresión).

7. **Cierre de pruebas y lecciones aprendidas:** Se elabora un informe consolidado de resultados, con evidencias documentales, validaciones realizadas, hallazgos resueltos y pendientes. Se documentan las lecciones aprendidas para mejorar futuros ciclos de prueba.

6.3 Técnicas y Herramientas Recomendadas

La ejecución eficaz de pruebas funcionales, técnicas y de seguridad en el marco del ciclo de vida de los sistemas de información requiere del uso de técnicas estructuradas y herramientas especializadas, preferiblemente integradas al entorno tecnológico institucional. A continuación, se presentan las principales técnicas y herramientas recomendadas para cada tipo de prueba, considerando la arquitectura en la nube basada en Microsoft Azure y el uso de tecnologías relacionadas en el documento Guía sobre el Framework Tecnológico para los Sistemas de Información Institucional.

6.3.1 Técnicas de Prueba

Funcionales	Pruebas de caja negra, validación de historias de usuario, pruebas exploratorias, pruebas automatizadas de UI.
Técnicas	Pruebas de carga, pruebas de rendimiento, pruebas de estrés, pruebas de compatibilidad, pruebas de regresión automatizadas
Seguridad	Análisis de vulnerabilidades, pruebas de penetración controladas (pentesting), validación de OWASP Top 10, pruebas de autenticación/autorización.

6.3.2 Herramientas de Apoyo

Tipo de Prueba	Herramienta Recomendada	Observaciones
Gestión de pruebas	Azure Test Plans (Azure DevOps)	Permite planificar, rastrear y ejecutar pruebas manuales y exploratorias. Se integra con pipelines de CI/CD.

Tipo de Prueba	Herramienta Recomendada	Observaciones
Pruebas automatizadas (UI / API)	Selenium / Postman	Selenium para pruebas web y Postman para validación de servicios RESTful, con posibilidad de integrar Newman en Azure Pipelines.
Pruebas de carga y rendimiento	Apache JMeter o Azure Load Testing	JMeter como estándar de la industria; Azure Load Testing permite pruebas escalables desde la nube.
Control de versiones y CI/CD	Azure Repos + Azure Pipelines	Facilitan la integración continua y despliegue automatizado de ambientes para pruebas.
Análisis de código estático	SonarCloud (integrado a Azure DevOps)	Evalúa calidad del código, cobertura de pruebas unitarias y detección temprana de vulnerabilidades.
Pruebas de seguridad (aplicaciones web)	OWASP ZAP, Burp Suite, Microsoft Defender for Cloud	OWASP ZAP y Burp Suite para pruebas manuales; Defender for Cloud para monitoreo continuo de vulnerabilidades en recursos Azure.
Registro de hallazgos y seguimiento	Azure Boards	Gestión de bugs, tareas de prueba y seguimiento del ciclo de pruebas asociado a los entregables del proyecto.

6.3.3 Integración con el Ciclo DevSecOps

Todas las herramientas seleccionadas deben alinearse con el modelo de entrega continua y segura, siguiendo prácticas DevSecOps. Se recomienda configurar pipelines en Azure DevOps que incluyan:

- Ejecución automática de pruebas unitarias y de integración al hacer push al repositorio.
- Evaluación de calidad del código.
- Pruebas automatizadas de API y UI tras cada despliegue a ambientes intermedios.
- Monitoreo de seguridad continuo con Microsoft Defender for DevOps.

¿Que son prácticas DevSecOps?	DevSecOps es la evolución de DevOps, y su propósito es garantizar que la seguridad no sea un paso posterior, sino un elemento transversal y automatizado durante todo el proceso de diseño, desarrollo, pruebas, despliegue y operación de las aplicaciones.

	Prácticas Recomendadas	Descripción
Principales Prácticas DevSecOps	Integración de análisis de seguridad en el código	Uso de herramientas de análisis estático (SAST) y dinámico (DAST) en el pipeline para detectar vulnerabilidades temprano.
	Pruebas automatizadas de seguridad	Ejecución de escaneos de seguridad en API, dependencias, contenedores y código cada vez que se despliega o actualiza.
	Gestión segura de secretos	Uso de herramientas como Azure Key Vault para manejar claves, tokens y contraseñas sin exponerlas en el código.
	Control de dependencias y librerías	Verificación automática de versiones vulnerables de librerías de terceros.

	Prácticas Recomendadas	Descripción
	Hardening de infraestructura como código (IaC)	Validación de las plantillas de despliegue (ARM templates, Bicep, Terraform) para evitar configuraciones inseguras.
	Monitoreo y respuesta activa	Uso de herramientas como Microsoft Defender for Cloud y Sentinel para detección, alertas y respuesta ante amenazas.
	Capacitación continua en seguridad para desarrolladores	Promover una cultura de desarrollo seguro dentro de los equipos de trabajo.

6.4 Roles y Responsabilidades

La correcta ejecución de las pruebas requiere de una asignación clara de responsabilidades entre los distintos actores involucrados en el desarrollo, despliegue y mantenimiento de los sistemas de información. A continuación, se detallan los roles clave y sus funciones específicas dentro del proceso de gestión de pruebas.

- **Jefe / Gerente de Proyecto**
  - Define el cronograma de pruebas dentro del plan general del proyecto.
  - Asegura que los recursos humanos, técnicos y de tiempo estén disponibles para la ejecución de las pruebas.
  - Supervisa el cumplimiento de las fases de prueba y la documentación de resultados.
  - Escala a instancias superiores cualquier bloqueo o hallazgo crítico que impida la continuidad del proceso.
- **Líder Técnico / Arquitecto de Solución**
  - Valida el entorno técnico para la ejecución de pruebas (ambientes, configuraciones, servicios integrados).
  - Asegura que los servicios en la nube y la infraestructura estén correctamente provisionados (por ejemplo, ambientes en Azure).
  - Define las rutas críticas de validación técnica y parámetros de calidad del sistema.
- **Ingeniero de Pruebas / Analista QA**
  - Diseña el plan de pruebas detallado, incluyendo los casos, escenarios y criterios de aceptación.
  - Ejecuta pruebas funcionales, técnicas y de seguridad, tanto manuales como automatizadas.
  - Registra los resultados y reporta hallazgos, errores o vulnerabilidades en la herramienta de gestión (Azure DevOps, Azure Boards).
  - Participa en la Re-ejecución de pruebas una vez corregidos los hallazgos por el equipo de desarrollo.
- **Equipo de Desarrollo (Backend / Frontend / Base de Datos)**
  - Corrige los errores detectados durante la ejecución de pruebas.
  - Realiza pruebas unitarias y de integración antes de liberar a ambientes de prueba formales.
  - Apoya la automatización de pruebas repetitivas, especialmente en servicios y componentes críticos.
  - Participa en sesiones de revisión técnica de los hallazgos junto con QA.
- **Equipo de Infraestructura**
  - Gestiona los entornos de pruebas y producción (provisión, respaldo, restauración).
  - Asegura que los entornos reflejen condiciones lo más cercanas posibles al entorno de operación.
  - Colabora en la ejecución de pruebas técnicas como pruebas de rendimiento, carga o stress.

- **Equipo de Seguridad de la Información**
  - Supervisa la aplicación de controles de seguridad durante las pruebas.
  - Define los requisitos de pruebas de seguridad y valida su ejecución conforme a normativas.
  - Evalúa hallazgos de seguridad y determina acciones de mitigación.
- **Área Usuaria / Funcional**
  - Participa activamente en la ejecución de pruebas de aceptación funcional (UAT).
  - Valida el cumplimiento de las funcionalidades frente a las historias de usuario aprobadas.
  - Emite la aprobación o rechazo del sistema desde el punto de vista operativo.
- **Administrador del Sistema / Soporte Técnico**
  - Brinda apoyo en el levantamiento de incidentes derivados del proceso de pruebas.
  - Realiza validaciones funcionales adicionales en representación de usuarios finales.
  - Apoya la estabilización del sistema pospruebas antes del paso a producción.

7. ENTREGABLES DE LA GESTIÓN DE PRUEBAS

Los entregables derivados del proceso de gestión de pruebas son fundamentales para evidenciar la validación técnica, funcional y de seguridad de las soluciones tecnológicas desarrolladas, adaptadas o adquiridas por la Agencia Nacional de Tierras (ANT). Estos documentos constituyen insumos clave para la toma de decisiones técnicas y administrativas, y respaldan el paso a producción de los sistemas de información.

A continuación, se describen los entregables mínimos requeridos para cada fase del proceso de pruebas:

- **Plan de Pruebas**
  - Documento que define el alcance, tipos de pruebas a ejecutar, cronograma, herramientas a utilizar, roles responsables y criterios de aceptación.
  - Debe estar alineado con el cronograma general del proyecto y aprobado por el líder técnico o el jefe de proyecto.

- **Casos y Escenarios de Prueba**
  - Matrices que detallan los pasos a ejecutar, condiciones iniciales, datos requeridos, entradas, resultados esperados y tipo de prueba.
  - Deben incluir pruebas positivas, negativas y pruebas críticas del negocio.
  - Pueden desarrollarse en herramientas como Azure Test Plans o en plantillas predefinidas por la ANT.

- **Registro de Evidencias**
  - Evidencias en formato digital (capturas de pantalla, logs, archivos exportados, etc.) que soporten los resultados de cada caso de prueba ejecutado.
  - Deben almacenarse organizadamente por módulo, fecha y tipo de prueba.
  - Pueden ser compartidas a través de Azure DevOps o repositorios oficiales habilitados por la Subdirección de Sistemas de Información de Tierras.

- **Informe de Resultados de Pruebas**
  - Documento que consolida los resultados obtenidos, casos aprobados, casos fallidos, observaciones, y hallazgos detectados.

- Debe contener: (1) Estado global del sistema probado (aprobado, aprobado con observaciones o rechazado). (2) Análisis de cumplimiento frente a los requerimientos funcionales y no funcionales. (3) Riesgos o recomendaciones derivados del proceso de prueba.

- **Registro de Hallazgos y Seguimiento**
  - Documento o tablero de control que permite rastrear errores, vulnerabilidades, desviaciones o incumplimientos identificados durante la ejecución.
  - Se debe indicar: (1) Descripción del hallazgo. (2) Severidad. (3) Estado (abierto, en análisis, corregido, validado). (3) Responsable de atención. (4) Fecha de cierre.

- **Acta de Aceptación de Pruebas (cuando aplique)**
  - Documento suscrito por el área funcional o el responsable del negocio que valida que el sistema cumple con los criterios establecidos y puede continuar al paso siguiente (homologación o producción).
  - Aplica especialmente en pruebas de aceptación de usuario (UAT).

- **Lecciones Aprendidas (opcional pero recomendado)**
  - Documento que recoge aspectos a mejorar para siguientes ciclos de prueba: errores recurrentes, deficiencias en el diseño, oportunidades de automatización, aspectos exitosos, entre otros.

Estos entregables deben ser archivados y referenciados en el expediente del proyecto o solución tecnológica, y estarán disponibles para fines de auditoría interna, control de calidad o revisión posterior por entes de control.

HISTORIAL DE CAMBIOS		
Fecha	**Versión**	**Descripción**
06/03/2026	1	Primera versión del documento. La estructuración de este instructivo es fundamental porque garantiza que todas las pruebas funcionales, técnicas y de seguridad se ejecuten de manera uniforme, controlada y trazable en los sistemas de información de la entidad. Sin un lineamiento claro, aumenta el riesgo de liberar software con fallas, vulnerabilidades, inconsistencias o incumplimientos normativos que afecten la operación misional. El documento estandariza procedimientos, define responsabilidades, asegura la calidad de los entregables y fortalece la confianza en los sistemas, reduciendo retrabajos y mejorando la continuidad y seguridad de las soluciones tecnológicas institucionales.

APROBACIÓN				
	NOMBRE	**CARGO**	**FIRMA**	**FECHA**
ELABORÓ	Jerson Eguis Berrio	Contratista - Subdirección de Sistemas de Información de Tierras	**ORIGINAL FIRMADO**	02/03/2026
REVISÓ	Liliam Cárdenas Diaz	Contratista - Subdirección de Sistemas de Información de Tierras	**ORIGINAL FIRMADO**	04/03/2026
APROBÓ	Diana Lucía Herrera Riaño	Subdirectora de Sistemas de Información de Tierras	**ORIGINAL FIRMADO**	06/03/2026