**TABLA DE CONTENIDO**

[1. INTRODUCCIÓN 2](#_Toc170463926)

[2. DESARROLLO 2](#_Toc170463927)

[2.1. Instrumentos y/o mecanismos 2](#_Toc170463929)

[2.1.1. Instrumento de recolección o acopio 2](#_Toc170463930)

[2.1.2. Documentos de la operación estadística (tipo parámetro) 3](#_Toc170463931)

[2.1.3. Herramientas tecnológicas 3](#_Toc170463932)

[2.1.4. Sensibilización 3](#_Toc170463933)

[2.1.5. Entrenamiento 3](#_Toc170463934)

[2.1.6. Procesamiento, análisis y difusión de información. 4](#_Toc170463935)

[2.2. Tipos de pruebas 5](#_Toc170463936)

[2.2.1. Prueba de escritorio 5](#_Toc170463937)

[2.2.2. Prueba cognitiva 5](#_Toc170463938)

[2.2.3. Prueba piloto 6](#_Toc170463939)

[2.2.4. Pruebas Funcionales y no Funcionales. 6](#_Toc170463940)

[3. ANEXOS 10](#_Toc170463942)

[4. CONTROL DE CAMBIOS 10](#_Toc170463943)

## INTRODUCCIÓN

Uno de los requerimientos del subsistema de gestión de la información estadística, es la realización de pruebas a los instrumentos y/o mecanismos que se diseñaron y construyeron durante las fases de diseño y construcción de un proceso estadístico para su implementación en una operación estadística en particular; por ello los resultados de las pruebas, junto con sus conclusiones deben quedar documentadas como evidencia del proceso estadístico correspondiente.

El presente documento se elaboró basado en la “*Guía para la definición y la aplicación de pruebas en las operaciones estadísticas*” (DANE, 2020, versión 2) y tiene como fin indicar las diferentes técnicas de pruebas que se pueden realizar, verificando así la funcionalidad de los instrumentos y/o mecanismos del proceso estadístico en una operación estadística, minimizando la presencia de errores y aportando al mejoramiento de la información estadística a difundir.

## DESARROLLO

En la presente guía se pueden observar cuatro tipologías de pruebas, como los son, prueba de escritorio, pruebas cognitivas o de entrevista, pruebas piloto y pruebas funcionales y no funcionales, sin embargo, la entidad puede apropiar otros tipos de prueba, las cuales puede documentarse en el formato vigente plan de pruebas DE\_FO\_26.

## Instrumentos y/o mecanismos

Para el desarrollo del proceso estadístico en una operación estadística, se debe contar con instrumentos y/o mecanismos diseñados que permitan la ejecución en sus diferentes fases y que susceptibles de ser puestos a prueba.

## Instrumento de recolección o acopio

Permiten la recolección o acopio de datos requeridos para dar cumplimiento al objetivo de la operación estadística.

Para este tipo de instrumentos, se deben tener en cuenta algunos de los siguientes criterios:

* Comprensión del instrumento (cuestionario o encuesta).
* Acceso al instrumento.
* Facilidad de diligenciamiento.
* Coherencia y consistencia en relación a la temática de la operación estadística.
* Secuencia de las preguntas (lógica).
* Redacción de las preguntas.
* Tiempo de diligenciamiento.

## Documentos de la operación estadística (tipo parámetro)

Documentos que enmarcan los lineamientos y actividades necesarios para la ejecución y comprensión de la operación estadística, dentro de los cuales están: los manuales, procedimientos, instructivos, guías, entre otros. Estos documentos deben surtir un proceso de validación donde se evalúan algunas de las siguientes características:

* Comprensión.
* Facilidad de uso.
* Integridad.
* Coherencia y consistencia en relación a la temática de la operación estadística.
* Redacción.
* Interacción con el Sistema de Gestión Integrado.

## Herramientas tecnológicas

La ejecución de pruebas al software aplicativo que se defina para el desarrollo de alguna fase o fases del proceso estadístico, las realiza el grupo GTIC como líder del Proceso de Gestión de Tecnologías y Seguridad de la Información, de acuerdo a los lineamientos diseñados para el desarrollo tecnológico.

## Sensibilización

Con el fin de que las partes interesadas conozcan el objetivo de la operación estadística y la información que produce, es necesario definir alternativas para llevar a cabo la trasmisión de la información. Algunos de los procesos de sensibilización a emplear, pueden ser:

* Piezas comunicativas (Banner, notas, videos, folletos, pancartas, entre otros).
* Presentaciones (virtuales o presenciales).
* Concursos (virtuales o presenciales).

## Entrenamiento

Actividad que implica la transmisión de conocimiento frente al rol en la fase de recolección o acopio de la Operación Estadística. Los elementos a someter a prueba pueden ser:

* La interpretación, la claridad y el dominio de los conceptos.
* Las técnicas de abordaje.
* La introducción a los encuestados sobre los objetivos de la operación.
* La apropiada respuesta de los encuestadores a las inquietudes de las fuentes.
* El manejo y el uso de los instrumentos.
* Los equipos de trabajo.
* La facilidad de ubicación en terreno.
* El seguimiento de instrucciones.
* La solución de situaciones planteadas
* Las habilidades en desarrollo de entrevistas.
* El liderazgo.

Asimismo, se pueden realizar pruebas a los siguientes procesos y actividades relacionados con el:

* Entrenamiento.
* La consecución de aulas para entrenamiento y condiciones de estas.
* La convocatoria y la recepción de hojas de vida.
* La preselección de aspirantes.
* La citación a entrenamiento.
* El manejo del material para entrenamiento.
* Los procesos y las estrategias de evaluación.
* La selección de personal.

## Procesamiento, análisis y difusión de información.

Las actividades que se realizan durante las fases de procesamiento, análisis y difusión de resultados son fundamentales para la depuración del archivo de datos, por lo que es necesario establecer las reglas y los desarrollos de software aplicativo apropiados para asegurar la consistencia de la información estadística que se pondrá a disposición de los usuarios.

Estas son las acciones posibles a efectuar:

* Verificar los desarrollos informáticos (para captura, edición e imputación, estimaciones, generación de cuadros de salida) con datos de prueba.
* Calcular tasas de imputación (cuando se aplica este procedimiento).
* Realizar controles de validación en los aplicativos de captura de los datos.
* Realizar controles de consistencia en los aplicativos de captura de los datos.
* Analizar resultados preliminares.

Cuando se realizan pruebas piloto o experimentales se deben llevar a cabo los procesos de captura, procesamiento de datos que incluye, tanto la edición y la imputación de datos como la producción de los cuadros de salida.

## Tipos de pruebas

Una vez seleccionado el instrumento o mecanismo diseñado y/o construido para el desarrollo de alguna actividad o fase de la operación estadística, y antes de su uso o puesta en producción, se debe someter a una prueba, con el fin de garantizar que el instrumento o mecanismo es funcional y no presentará errores durante su aplicación o puesta en producción. Para ello, a continuación se presentan algunos tipos de pruebas que se pueden aplicar.

## Prueba de escritorio

Esta prueba consiste en realizar una verificación a los instrumentos y/o mecanismos seleccionados, desde la experiencia en la temática a tratar. Para ello, es fundamental que el personal que está directamente relacionado con la temática de la operación estadística las realice, o por el contrario si es verificar la metodología del instrumento frente a un lineamiento interno o externo de construcción o estructuración de documentos tipo parámetro, debe contar con experiencia relacionada con el tema.

En esta prueba, generalmente se realizan las siguientes verificaciones, visto desde lo temático y lo metodológico:

* Redacción de preguntas (para el caso de encuestas).
* Comprensión del texto y contenidos, los cuales deben estar relacionados con la temática de la operación estadística, pero, en un lenguaje claro y entendible puesto que están dirigidos a usuarios en general.
* Secuencia lógica y cumplimiento de lineamientos internos o externos.
* Facilidad de uso.
* Funcionalidad

Para documentar los resultados de este tipo de prueba, no es necesario usar el formato vigente plan de pruebas E1-FO-03, se puede usar el correo electrónico junto con los documentos virtuales y/o reuniones virtuales o presenciales para realizar los aportes, recomendaciones, observaciones, ajustes y/o verificaciones del caso, siguiendo lo contemplado en el procedimiento vigente de control de documentos del Sistema de Gestión Integrado E1-PR-01 y el Instructivo vigente elaboración, actualización y derogación de documentos del Sistema de Gestión Integrado -SGI E1-IN-01 para los casos que haya lugar.

## Prueba cognitiva

Este tipo de pruebas se puede dividir en dos formas: A través de entrevista o cuestionario, y está dirigida principalmente a evaluar los instrumentos de recolección/acopio, difusión y encuestas.

A través de la formulación de preguntas se busca conocer la percepción que tienen los entrevistados o encuestados del instrumento de recolección/acopio o del diseño de una encuesta en prueba. Se deben seleccionar las personas que se encargarán de diligenciar inicialmente el instrumento de recolección o encuesta que se desea probar, y posteriormente contestarán un cuestionario diseñado por el líder temático o quien se designe. Las preguntas a formular en este cuestionario pueden estar enfocadas en los siguientes aspectos:

* Ingreso o acceso al instrumento de recolección/acopio o encuesta.
* Comprensión del instrumento de recolección/acopio o preguntas de la encuesta.
* Facilidad en su diligenciamiento.
* Tiempo de diligenciamiento.
* Facilidad de uso.
* Pregunta abierta frente a que puede mejorar desde su punto de vista.

Para documentar la metodología a seguir, se debe usar el formato vigente plan de pruebas E1-FO-03. Una vez concluida la prueba, se debe documentar los resultados en el formato vigente denominado “*informe de pruebas*” con código E1-FO-02.

## Prueba piloto

Generalmente, la aplicación de esta prueba está enfocada a la realización de las fases del proceso estadístico, desde la recolección o acopio de la información hasta la difusión de la información estadística antes de su puesta en marcha. La idea es poder evaluar los instrumentos y mecanismos, como lo es el instrumento de recolección o acopio, el o los sistemas de información que intervienen y el entrenamiento del personal de la operación estadística.

Esta prueba pretende verificar lo diseñado o construido, frente a una simulación real de la operación estadística, pudiendo encontrar errores, inconsistencias o falta de entrenamiento al momento de ejecutarla. También se debe contar con una muestra representativa de las fuentes de información del proceso.

Para documentar la metodología a seguir, se puede usar el formato vigente plan de pruebas E1-FO-03 y documentar los resultados en el formato vigente informe de pruebas E1-FO-02. Adicionalmente, los ajustes y mejoras del software aplicativo en relación con las fases de acopio, procesamiento, análisis y difusión se pueden hacer uso de los GLPI´s, herramienta técnica para administrar el servicio de asistencia técnica y que evidencia el requerimiento solicitado, fecha de implementación y fecha de validación por parte del solicitante.

## Pruebas Funcionales y no Funcionales.

En el ámbito del desarrollo de software, la implementación de un robusto protocolo de pruebas es crucial para asegurar la calidad y funcionalidad del producto final. Este documento detalla las prácticas recomendadas y las herramientas necesarias para establecer un enfoque integral de pruebas, desde pruebas unitarias hasta pruebas de aceptación.

Para el ambiente específico del aplicativo RUNAP se realizarán 3 tipos de pruebas

1. Unitarias
2. Cypress
3. Integración y
4. Aceptación
5. **Las pruebas unitarias:** son esenciales para verificar la correcta ejecución de los componentes más pequeños de un proyecto. Estas pruebas se centran en funciones o métodos individuales para garantizar que responden como se espera en diversos escenarios. PHPUnit es un framework de pruebas unitarias para PHP, ampliamente utilizado para garantizar que el código PHP funcione como se espera. A continuación, se muestra un ejemplo básico y algunas mejores prácticas asociadas con PHPUnit.

* **Definición y propósito**: Las pruebas unitarias se diseñan para validar cada componente individual del código, asegurando que se comporta correctamente de manera aislada. Su principal propósito es identificar errores a nivel de desarrollo temprano, lo que ayuda a evitar complicaciones en etapas más avanzadas del ciclo de vida del software.
* **Herramientas y marcos comunes**: Herramientas como JUnit para Java, PyTest para Python, y frameworks como NUnit para .NET son ampliamente utilizados para la implementación de pruebas unitarias. Estos marcos ofrecen funcionalidades como aserciones, setup y teardown para configurar el ambiente de pruebas, y la ejecución de suites de pruebas.
* **Mejores prácticas**: Es crucial mantener las pruebas unitarias rápidas y enfocadas. Deben ser independientes de otras pruebas y capaces de ejecutarse en paralelo para maximizar la eficiencia. La cobertura del código es también un indicador clave de la efectividad de las pruebas unitarias.
* **Pruebas Funcionales - GLPI’s**: La implementación de mejoras en el software aplicativo pueden ser solicitadas mediante el uso de los GLPI.

**Las mejores Prácticas para PHPUnit**

* **Escribir pruebas limpias y descriptivas:** Cada función de prueba debe centrarse en un solo aspecto del comportamiento del componente bajo prueba. Utiliza nombres descriptivos para las funciones de prueba, que expliquen claramente lo que están verificando.
* **Aislamiento de pruebas:** Las pruebas no deben depender unas de otras. Cada prueba debe ser capaz de ejecutarse de forma independiente y en cualquier orden.
* **Utilizar datos de prueba:** PHPUnit permite utilizar proveedores de datos (@dataProvider) para ejecutar una función de prueba varias veces con diferentes entradas, lo que facilita la cobertura exhaustiva de los casos de uso.

Ejemplo con **@dataProvider**:

phpCopy code

class SumaTest extends TestCase {

/\*\* \* @dataProvider sumaProvider \*/

public function testSuma($a, $b, $expected) {

$this->assertEquals($expected, suma($a, $b));

}

public function sumaProvider() {

return [ [0, 0, 0], [1, 1, 2], [2, 2, 4], [-1, -1, -2], ];

}

}

* **Simular dependencias:** A menudo, las unidades de código dependen de otras clases o funciones. PHPUnit ofrece mock objects y stubs para simular estas dependencias, permitiendo que las pruebas se centren en el código bajo prueba sin preocuparse por las dependencias externas.
* **Ejecución de pruebas y cobertura de código:** Utiliza la línea de comandos para ejecutar tus pruebas regularmente y considera la integración de estas ejecuciones en tu entorno de integración continua (CI). La generación de informes de cobertura de código también es crucial para identificar áreas del código que no están siendo probadas.
* **Manejo de excepciones y errores:** Es importante probar cómo el código maneja situaciones inesperadas, como entradas inválidas o fallas en las dependencias. PHPUnit permite verificar si se lanzan excepciones correctamente bajo ciertas condiciones.

Ejemplo de prueba de excepciones:

phpCopy code

public function testException() {

$this->expectException(InvalidArgumentException::class); someFunctionThatThrowsAnExceptionWithInvalidArgument();

}

Estas prácticas y configuraciones ayudarán a los desarrolladores a maximizar los beneficios de usar PHPUnit en sus proyectos, asegurando que las pruebas sean robustas, confiables y mantenibles.

1. **Pruebas con Cypress:** es una herramienta avanzada para la realización de pruebas de interfaz de usuario que simula la interacción real del usuario con una aplicación web. A continuación, se muestra cómo configurar una prueba básica para verificar si un botón de inicio de sesión lleva correctamente a la página de inicio después de un clic.

* **Introducción a Cypress:** Cypress ofrece una nueva aproximación a las pruebas de interfaz de usuario realizadas directamente en el navegador. A diferencia de otras herramientas como Selenium, Cypress se ejecuta en el mismo bucle de ejecución que la aplicación, ofreciendo mayor velocidad y fiabilidad.
* **Configuración y casos de uso:** La configuración de Cypress es sencilla, con soporte para JavaScript y frameworks frontend populares como React y Angular. Es especialmente útil para realizar pruebas en aplicaciones de página única (SPA).
* **Ventajas sobre otras herramientas de pruebas de interfaz de usuario**: Cypress incluye funcionalidades automáticas de espera, no utiliza Selenium y tiene una arquitectura más moderna que reduce el tiempo y el esfuerzo en la configuración y ejecución de pruebas.

describe('Prueba de inicio de sesión', () => {  
 it('debe iniciar sesión exitosamente', () => {  
 cy.visit('https://example.com/login');  
 cy.get('input[name=username]').type('usuario');  
 cy.get('input[name=password]').type('contraseña');  
 cy.get('form').submit();  
 cy.url().should('include', '/dashboard');  
 });  
});

1. **Pruebas de Integración:** Las pruebas de integración son cruciales en el proceso de desarrollo de software porque aseguran que los componentes individuales (previamente verificados mediante pruebas unitarias) funcionen juntos como se espera. Estas pruebas son fundamentales para detectar problemas en las interfaces y en la interacción entre módulos, sistemas o servicios que pueden no ser aparentes en las pruebas unitarias.

* **Importancia de las Pruebas de Integración:** En las pruebas de integración, los componentes son combinados y probados como un grupo. Estas pruebas pueden revelar problemas en la interfaz y en las interacciones entre componentes, incluyendo errores de comunicación, de interfaz, de datos compartidos, y de comportamiento bajo carga.

**Metodologías Comunes**

* **Integración Big Bang:** Este método implica integrar todos los componentes de una vez y probar el conjunto entero. Aunque es simple, puede ser difícil identificar la fuente de un error debido a la cantidad de componentes involucrados.
* **Integración Incremental:** Preferida por muchos, esta metodología implica integrar dos o más componentes que ya han sido probados individualmente, verificar su funcionalidad juntos, y luego agregar otros componentes progresivamente. Dentro de esta categoría, existen dos enfoques:
  + **Integración Top-Down**: Se inicia con los componentes de nivel superior y gradualmente se integran los componentes de nivel inferior.
  + **Integración Bottom-Up**: Se comienza con los componentes de bajo nivel, y gradualmente se integran hacia los de nivel superior.

**Pruebas de sistema:** Las pruebas de sistema involucran la validación completa del sistema en entornos que simulan la producción. Estas pruebas son integrales y cubren todas las funciones y características del software.

* **Cómo abordar las pruebas de sistema:** Deben realizarse en un entorno que imite al entorno de producción lo más cercanamente posible, utilizando datos de prueba que reflejen los datos reales de operación.
* **Herramientas recomendadas:** Herramientas como LoadRunner y JMeter son utilizadas frecuentemente para pruebas de carga, mientras que Selenium se utiliza para pruebas funcionales automatizadas a nivel de sistema.

1. **Pruebas de aceptación:** Estas pruebas verifican si el sistema cumple con los requisitos del negocio y si es aceptable para el usuario final. Son la última fase antes de la entrega del software al cliente.

* **Definición y relevancia:** Las pruebas de aceptación son realizadas por usuarios finales y verifican tanto la funcionalidad como la usabilidad del software en situaciones reales de uso.
* **Proceso de ejecución de pruebas de aceptación:** Generalmente se llevan a cabo en un entorno de "pre-producción", utilizando escenarios de prueba basados en las especificaciones originales del sistema. Herramientas como Cucumber son populares para estas pruebas por su enfoque en el comportamiento.

## ANEXOS

No Aplica.

## CONTROL DE CAMBIOS

| **FECHA DE VIGENCIA VERSIÓN ANTERIOR** | **VERSIÓN ANTERIOR** | **MOTIVO DE LA ACTUALIZACIÓN** |
| --- | --- | --- |
| 19/12/2023 | 1 | Se re codifica el documento de acuerdo con el nuevo mapa de procesos, actualizando el código. El documento por cargue inicial en la aplicación tecnológica reinicia desde el código 1. Para consultar los obsoletos ver matriz de armonización documentos del SGI al nuevo mapa de procesos <https://drive.google.com/drive/u/1/folders/1Tu2ChzlvgSaXxc10UpqzX-SVhu095Kvv>  Las fechas que aparecen en el control de revisión y aprobación, obedecen a las fechas registradas en el documento antes de la migración del documento al nuevo mapa de procesos. |
| 28/06/2024 | 2 | Se incluyen las pruebas funcionales y no funcionales al software aplicativo, dando alcance al procedimiento “Pruebas y evaluación de los mecanismos e instrumentos del proceso estadístico” y se incluyen los GLPI como mecanismo de prueba al software aplicativo, dando cierre a la acción correctiva de la no conformidad 11 del plan de mejoramiento suscrito con el DANE, derivado de la evaluación realizada al proceso estadístico implementado en la operación estadística "Áreas Protegidas Integrantes del SINAP, Inscritas en el RUNAP". |

|  |  |  |
| --- | --- | --- |
| **CRÉDITOS** | | |
| Elaboró / Actualizó | Nombre | Jairo González - Marcela Alvear  Alan Aguía  Marcela Borda |
| Cargo | Contratistas Grupo de Gestión e Integración del SINAP  Contratista Grupo de Tecnologías de la Información y Comunicaciones  Contratista Oficina Asesora de Planeación |
| Fecha | 28/06/2024 |
| Revisó | Nombre | Luis Alberto Cruz Colorado |
| Cargo | Coordinador Grupo de Gestión e Integración del SINAP |
| Fecha: | 28/06/2024 |
| Aprobó | Nombre | Andrés León |
| Cargo | Jefe Oficina Asesora de Planeación |
| Fecha: | 28/06/2024 |