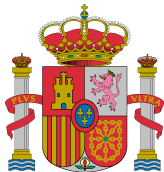


# Guía de elaboración planes de pruebas

Oficina de Calidad

Versión: 2.4.0


 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD  SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD
--	---------------------------------------	---

## Control de versiones


Versión	Fecha	Autor	Descripción / Comentarios
1.00	02-10-2015	SGTIC	Primera versión
2.00	13-01-2015	SGTIC	Reestructuración completa del documento original. Se amplían y detallan los niveles de prueba recogidos en el Plan de Pruebas. Se identifican y describen las herramientas establecidas por el Área de Desarrollo de la SGSICS para la consecución de los objetivos de calidad del software marcados.[borrador]
2.01	03-02-2015	SGTIC	Documento liberado tras revisión realizada por parte de los responsables del Área de Desarrollo.
00	21-04-2016	SGTIC	Publicar
	16/06/2016	SGTIC	Revisión V01_v00 para publicar y se quita la versión específica de la carátula, dejando solo el texto Versión
V02_v02	30/10/2017	Oficina Calidad	Revisión: Cambios en cabecera, versiones de las herramientas, y SGSICS por SGSICS
V02_v03	29/11/2018	Oficina Calidad	Revisión: Eliminación de anexos y duplicidades con documentos de normativa.
2.4.0	02/03/2020	Oficina Calidad	Se añade los apartados de Pruebas de Regresión y Pruebas de API o Contrato (MicroServicios)

## Índice

1. OBJETO.....	5
2. INTRODUCCIÓN .....	5
2.1. ALCANCE .....	6
2.2. GLOSARIO.....	6
2.3. DOCUMENTOS DE REFERENCIA.....	6
3. FASE DE ANÁLISIS FUNCIONAL DEL SISTEMA: ESPECIFICACIÓN DEL PLAN DE PRUEBAS.....	7
3.1. INTRODUCCIÓN DEL SISTEMA .....	7
3.2. DEFINICIÓN DEL ALCANCE Y ESTRATEGIA DE PRUEBAS.....	7
3.2.1. Pruebas Funcionales .....	7
3.2.2. Pruebas no Funcionales .....	8
3.2.3. Pruebas Unitarias .....	8
3.2.4. Pruebas de Rendimiento .....	8
3.2.5. Pruebas de Regresión .....	10
3.2.6. Pruebas de API o Contrato (Solo MicroServicios).....	10
3.3. DEFINICIÓN ESCENARIOS Y ENTORNOS DE PRUEBAS.....	10
3.3.1. Entornos .....	10
3.3.2. Herramientas .....	12
3.4. DEFINICIÓN DEL PLAN DE PRUEBAS DE ACEPTACIÓN DEL SISTEMA.....	12
4. FASE DE DISEÑO TÉCNICO DEL SISTEMA: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	13
4.1. ESPECIFICACIÓN DEL ENTORNO DE PRUEBAS .....	13
4.1.1. Plataforma de Gestión de Pruebas: TestLink.....	13
4.1.2. Framework para Pruebas unitarias: JUnit .....	14
4.1.3. Framework de Pruebas automatizadas: Selenium .....	14
4.1.4. Herramienta para pruebas de servicios web: SoapUI .....	14
4.1.5. Herramienta para pruebas de rendimiento y escalabilidad: JMeter .....	14
4.1.6. Herramientas para pruebas de regresión:.....	14
4.1.7. Herramientas para pruebas de API o Contrato: .....	15
4.2. ESPECIFICACIÓN TÉCNICA DE NIVELES DE PRUEBA.....	15
4.2.1. Especificación técnica Pruebas de Rendimiento.....	16
4.3. REVISIÓN DE LA PLANIFICACIÓN DE PRUEBAS .....	16
5. FASE DE CONSTRUCCIÓN Y EJECUCIÓN DE LAS PRUEBAS.....	17
5.1.1. Pruebas unitarias (JUnit) .....	17

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

5.1.2.	Pruebas funcionales automatizadas (Selenium).....	17
5.1.3.	Pruebas funcionales de servicios web (SoapUI) .....	17
5.1.4.	Pruebas de rendimiento (JMeter) .....	17
5.1.5.	Pruebas de Regresión .....	17
5.1.6.	Pruebas de API o Contrato (Solo MicroServicios) .....	18
6.	FASE DE ACEPTACIÓN DEL SISTEMA.....	18

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

## 1. OBJETO

El objeto de este documento es la definición de la metodología a seguir para la definición de los Planes de Pruebas de los Sistemas de Información desarrollados para la SGSICS en el marco de la Gestión de la Calidad del software.

## 2. INTRODUCCIÓN

El plan de pruebas permite certificar el grado de adecuación y, por tanto, la viabilidad de la puesta en explotación de un proyecto o entrega. El plan de pruebas define los criterios e indicadores mínimos exigibles para una evaluación objetiva de los sistemas de información a implementar.

El presente documento recoge:

- La definición de entornos, herramientas y procedimientos de ejecución de pruebas
- El procedimiento de definición de los catálogos de pruebas a realizar sobre el sistema.
- Los requisitos relativos a la planificación de las pruebas

El plan de pruebas resultante deberá identificar las pruebas a realizar sobre los distintos elementos y funcionalidades del sistema, los procedimientos de ejecución de las pruebas, los resultados esperados y la planificación global de las pruebas.

El Plan de Pruebas General podrá incluir los siguientes tipos de pruebas.

- Pruebas unitarias (automatizadas):
  - o Pruebas de caja blanca
  - o Pruebas de caja negra
- Pruebas de integración
- Pruebas de sistema
- Pruebas de implantación
- Pruebas funcionales (automatizadas)
- Pruebas no funcionales:
  - o Pruebas de rendimiento
  - o Pruebas de carga y estabilidad
  - o Pruebas de capacidad o estrés
  - o Prueba de escalabilidad
- Pruebas de aceptación

El conjunto de pruebas a realizar dependerá de las características del sistema a implementar. Estas deberán ser determinadas en la fase de análisis.

Este documento toma como punto de partida la fase de Análisis Funcional y va identificando los productos de entrada y salida de cada una de las fases del ciclo de vida de implementación del Sistema (Diseño, Construcción e Implantación) incorporando en cada una de las fases la planificación de las pruebas que deben diseñarse en las mismas.

## 2.1. ALCANCE

La normativa detallada en el presente documento es de obligado cumplimiento para todas aquellas personas, equipos y empresas responsables de la definición, implementación, ejecución y evaluación de resultados de los Planes de Prueba requeridos dentro del marco de aseguramiento de la calidad de los sistemas de información desarrollados para la SGSICS.

## 2.2. GLOSARIO

Término	Descripción
<b>Requisito funcional</b>	Define una función o comportamiento del sistema de software o de sus componentes. En la metodología, los requisitos funcionales se consideran de tipo casos de uso.
<b>Requisito no funcional</b>	Requisito que especifica criterios que pueden usarse para <b>juzgar la operación de un sistema</b> en lugar de sus comportamientos específicos, los cuales están determinados por los requisitos funcionales. No describen información a guardar, ni funciones a realizar, sino características de funcionamiento.  Requisitos no funcionales típicos: Rendimiento, escalabilidad, estabilidad, interoperabilidad, seguridad, usabilidad.
<b>Caso de uso (use case)</b>	Descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso (requisito funcional). Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema
<b>Caso de prueba (test case)</b>	Conjunto de condiciones o variables bajo las cuales se determina si una aplicación, un sistema software, o una característica de éstos es parcial o completamente satisfactoria.

## 2.3. DOCUMENTOS DE REFERENCIA

/Ubicación/Documento	Descripción
Cronograma del Proyecto	
PublicacionIntranet/plantillas/doc/01.ASI/MIR-APLIC-ASI-	Entregable de Especificaciones

ESPECIFICACIONES.docx	
PublicacionIntranet/22-PRUEBAS/MIR-NORMA-PrFuncAutJava.docx	Normativa para el desarrollo de las pruebas funcionales automáticas de las aplicaciones java y su ejecución.
PublicacionIntranet/22-PRUEBAS/MIR-NORMA-PrRendAut.docx	Normativa para la creación de los scripts de las pruebas no funcionales.

### 3. **FASE DE ANÁLISIS FUNCIONAL DEL SISTEMA: ESPECIFICACIÓN DEL PLAN DE PRUEBAS**

En la fase inicial de análisis funcional del sistema debe identificarse y documentarse, dentro de la documentación del **Plan de Pruebas**, los puntos identificados a continuación

#### 3.1. INTRODUCCIÓN DEL SISTEMA

En este punto se detallarán las siguientes cuestiones.

- *Breve descripción de la aplicación y las funcionalidades que satisface*
- *Modos de acceso a la aplicación y escenarios de uso*
- *Dimensionamiento en relación con el volumen de uso (conurrencia de usuarios) y de datos esperado.*

#### 3.2. DEFINICIÓN DEL ALCANCE Y ESTRATEGIA DE PRUEBAS

El alcance de las pruebas dependerá de la tipología, criticidad y tecnología del sistema a implementar por lo que dicho alcance deberá ser definido durante la **fase de análisis del sistema** partiendo de los **requisitos o especificaciones recogidas en los pliegos** a este respecto.


En este punto el Plan de Pruebas deberá:

- *Explicar la estrategia o estrategias a seguir para una prueba completa de la aplicación identificando los tipos o niveles de pruebas que serán implementados.*
- *Señalar los puntos críticos más importantes de las pruebas*

A continuación, se describen los diferentes tipos de pruebas que permitirán probar los sistemas de información desarrollados.

##### 3.2.1. **Pruebas Funcionales**

Las **pruebas funcionales** están orientadas a asegurar que se cumplen los **requisitos funcionales definidos por el usuario** y que el sistema está libre de errores. Son pruebas que se llevan a cabo a través de la interfaz gráfica del software (GUI) y su resultado impacta directamente en la percepción del usuario respecto a la calidad del sistema. Estas pruebas han de demostrar que las funcionalidades del sistema son operativas, que la entrada y salida de información en el sistema se produce de forma correcta y que se mantiene la integridad de esta.

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

Las **pruebas funcionales** se han de **automatizar** para permitir la reproducción de las mismas de forma ágil permitiendo realizar pruebas de regresión del sistema. Esto es, comprobar que nada de lo probado con anterioridad ha dejado de funcionar tras la modificación del sistema.

### 3.2.2. Pruebas no Funcionales

Las **pruebas no funcionales** se centran en los aspectos del software que no están relacionados directamente con una acción del usuario o una función específica del mismo, tales como el rendimiento, la escalabilidad, la seguridad, etc.

Su propósito es verificar la correcta operación y funcionamiento de los aspectos del sistema que no son directamente perceptibles por el usuario, aunque si entran dentro de sus expectativas.

### 3.2.3. Pruebas Unitarias

Las pruebas unitarias permiten comprobar el correcto funcionamiento de cada componente o módulo del sistema de información de forma individual. Tienen como objetivo verificar la funcionalidad y estructura de cada componente individual asegurando el correcto funcionamiento de las interfaces y el flujo de datos entre componentes.

Dentro de las pruebas unitarias se distinguen dos tipos de pruebas.

#### 3.2.3.1 Pruebas de caja blanca

Las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. El probador (tester) escoge distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados. Las pruebas de caja blanca son aplicables a varios niveles pero habitualmente se aplican a las unidades de software

#### 3.2.3.2 Pruebas de caja negra

Las pruebas de caja negra son aquellas que prueban el software desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno


### 3.2.4. Pruebas de Rendimiento

Las **pruebas de rendimiento** son pruebas no funcionales cuyo propósito es verificar el comportamiento del sistema bajo diferentes escenarios de carga de trabajo.

En lo relativo a las **pruebas de rendimiento** del sistema, se deberá realizar un análisis del sistema a implementar para identificar los diferentes escenarios de carga que se deberán simular en la fase de pruebas. Las pruebas de rendimiento se harán necesarias siempre que el sistema presente alguna de las siguientes características:

- El volumen de usuarios sea elevado.
- El sistema implemente procesos de negocio de alta demanda.
- El sistema implemente procesos de alta demanda de información.
- El sistema implemente procesos costosos.



 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD  SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD
--	---------------------------------------	---

- El sistema tenga de proporcionar información en un tiempo de respuesta reducido.
- El sistema tenga que almacenar una cantidad significativa de información en un tiempo reducido.
- El sistema implemente servicios que han de ser consumidos por otros sistemas.

Las pruebas de rendimiento a realizar serán las siguientes.

#### 3.2.4.1 Pruebas de carga y estabilidad

El objetivo de las **pruebas de carga** es verificar que el sistema es capaz de soportar el volumen de peticiones (carga) esperado (requerido) sin presentar degradación en el rendimiento del mismo.

Esta carga puede ser el número esperado de usuarios concurrentes que utilizarán la aplicación, realizando un número específico de transacciones durante un período de tiempo establecido.

Las **pruebas de estabilidad** determinan si la aplicación puede aguantar una carga esperada **continuada** de trabajo.

El escenario de las pruebas se diseñará estableciendo una duración de 2 horas durante las cuales el sistema deberá soportar el número de usuarios concurrentes y ritmo transaccional definidos en los requisitos de rendimiento con objeto de verificar la **estabilidad** del mismo.

#### 3.2.4.2 Pruebas de capacidad o estrés

Pruebas utilizadas para determinar los límites de carga que puede soportar un sistema con los recursos asignados para la carga inicialmente esperada. Estas pruebas permiten identificar aquellas partes del sistema que suponen un cuello de botella, así como el punto en el que el sistema se vuelve inutilizable.

El número de usuarios ha de ser incrementado en progresión geométrica hasta alcanzar el nivel de saturación del sistema en el cual los tiempos de respuesta quintuplican los límites establecidos o se produce una denegación de servicio en un porcentaje de peticiones superior al 5%.

#### 3.2.4.3 Pruebas de escalabilidad

La escalabilidad se entiende como la capacidad que tiene el sistema para, sin aplicar cambios drásticos en su configuración, soportar el incremento de demanda en la operación.

Las pruebas de escalabilidad permiten, por tanto, verificar que el sistema es capaz de soportar una mayor carga cuando se amplían los recursos asignados al mismo.


Tomando como referencia los resultados obtenidos en la prueba de capacidad o estrés se deberá verificar que el sistema vuelve a proporcionar el rendimiento esperado tras ampliar los recursos asignados al mismo. Esta ampliación de recursos podrá tener lugar en dos direcciones:

-Escalado vertical: Ej. Incremento en la memoria RAM o CPU's en su infraestructura.

-Escalado horizontal: Ej. Agregación de un nodo extra en un esquema de balanceo en la capa aplicativa o de base de datos

#### 3.2.4.4 Pruebas de Aceptación

Las **pruebas de aceptación**, involucran al usuario final y pretenden comprobar que la solución cumple los requisitos básicos de funcionamiento esperado y permiten que el usuario determine la

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

aceptación del sistema, la detección de defectos del producto entregado y planes de acción para corrección de los mismos.

### 3.2.5. Pruebas de Regresión

Hay que tener presente que cuando se realiza cualquier cambio en el código de una aplicación, por muy pequeño o muy simple, puede tener consecuencias inesperadas.

Estos cambios, pueden romper, dar unos comportamientos distintos a los esperados en partes del código que antes funcionaban y, además, en partes de código que pensamos que no estaban implicadas o relacionadas con la modificación realizada. Incluso pueden aparecer errores que ya se habían solucionado.

Un pequeño cambio puede “Invalidar” partes del código que antes funcionaban y que parecía que no tenían relación alguna con la modificación que hemos realizado. Incluso puede hacer que errores que ya hemos solucionado vuelvan a aparecer, por mencionar algunos ejemplos.

### 3.2.6. Pruebas de API o Contrato (Solo MicroServicios)

Las pruebas de API o Contrato, aparecen con el paradigma de los microservicios. Los microservicios lo definió Martin Fowler en 2014 como un enfoque técnico para desarrollar aplicaciones como un conjunto de pequeños servicios. Estos pequeños servicios serían unidades funcionales independientes que se ejecutan de manera autónoma en su propio proceso y que se comunican entre sí a través de APIs (interfaces).

Teniendo en cuenta, que la parte importante de un microservicio es su comunicación entre los diferentes servicios, las pruebas de estos se centraran en sus APIs, para garantizar que la comunicación se realice correctamente.

Además, este tipo de pruebas nos permiten verificar los esquemas de solicitud y de respuesta de una API y su comportamiento, es decir, qué información retorna un endpoint con diferentes solicitudes.

## 3.3. DEFINICIÓN ESCENARIOS Y ENTORNOS DE PRUEBAS


### 3.3.1. Entornos

En la actualidad la SGSICS dispone de los siguientes entornos:

- Desarrollo
- Pre-producción
- <Producción>

No se dispone de entornos específicos para la ejecución de los distintos tipos de pruebas (integración, rendimiento, etc.). Por tanto, las pruebas serán ejecutadas en los entornos de desarrollo o pre-producción en función de los condicionantes existentes en cada situación.

En la documentación de **Plan de Pruebas** deberá **incluirse información relativa a los entornos de pruebas asignados para cada una de las pruebas**.

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

El **entorno de desarrollo** está a completa disposición de los **proveedores externos** que desarrollan el sistema de información en cuestión de modo que estos pueden desarrollar, desplegar y ejecutar las pruebas sobre dicho entorno. No obstante, la certificación final de las pruebas será llevada a cabo sobre el entorno de **pre-producción** garantizando esto la portabilidad de dichas pruebas.

Dado que todos los sistemas de información están replicados en el entorno de **desarrollo**, este entorno puede ser considerado por el proveedor como un entorno válido para sus **pruebas de integración de sistemas**, si bien las pruebas de integración vinculantes serán aquellas que se realicen sobre el entorno de **pre-producción**.

Del mismo modo las **pruebas de aceptación del usuario** se llevarán a cabo sobre el entorno de **pre-producción**.

Las **pruebas unitarias de código** se consideran pruebas estrechamente ligadas al **desarrollo** por lo que el ambiente de ejecución es la plataforma.

En este punto del Plan de Pruebas se deberán identificar los entornos que van a ser utilizados para la ejecución de las pruebas en función de los condicionantes propios de cada proyecto.

Elemento	Descripción	Prioridad (Alta, media, baja)
Contexto de las pruebas	Detallar la configuración de equipos y otros dispositivos necesaria para hacer las pruebas	<alta, media, baja>
Modos de acceso a la aplicación y escenarios de uso que deben ser probados	<Detallar el modo de acceso a la aplicación a probar (acceso web, web services) >	<alta, media, baja>
Enumerar sistemas que deben estar disponibles y/o accesibles para las pruebas	<enumerar sistemas>	<alta, media, baja>
Otros prerequisites para poder realizar las pruebas	<Detallar otros prerequisites>	<alta, media, baja>
Identificar los scripts y los documentos para los Juegos de datos necesarios para hacer las pruebas	<Identificar los juegos de datos para realizar las pruebas >	<alta, media, baja>

### 3.3.2. Herramientas


A continuación, se enumeran las herramientas seleccionadas por la SGSICS para el soporte de los Planes de Pruebas de los sistemas de información implementados.

- Plataforma de Gestión de Pruebas: TestLink
- Framework para Pruebas unitarias: JUnit
- Framework de Pruebas automatizadas: Selenium
- Herramienta para pruebas de servicios web: SoapUI
- Herramienta para pruebas de rendimiento y escalabilidad: JMeter
- Herramientas para pruebas de regresión: Selenium, JMeter y SoapUI
- Herramientas para pruebas de API o Contrato: En estudio

### 3.4. DEFINICIÓN DEL PLAN DE PRUEBAS DE ACEPTACIÓN DEL SISTEMA

El Plan de Aceptación será el documento de referencia que permitirá al usuario validar el cumplimiento de los requisitos acordados como último paso previo a la puesta en explotación.

El Plan de Pruebas de Aceptación toma como referencia el documento de **especificación de requisitos del usuario** y define el conjunto de pruebas que validarán si el sistema cumple con el funcionamiento esperado. El conjunto de pruebas identificado en el Plan de Pruebas de aceptación debe asegurar que se cumplen tanto los requisitos funcionales especificados por el

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD  SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD
--	---------------------------------------	---

usuario como los requisitos no funcionales identificados durante el análisis del sistema (rendimiento, seguridad, disponibilidad, etc.).

#### 4. **FASE DE DISEÑO TÉCNICO DEL SISTEMA: ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS**

En esta actividad se realiza la **especificación del detalle del plan de pruebas** del sistema de información para cada uno de los niveles de prueba establecidos en el proceso **Análisis del Sistema de Información (ASI)**:

- Pruebas funcionales (automatizadas):
- Pruebas no funcionales:
  - Pruebas de integración
  - Pruebas de sistema
  - Pruebas de implantación
    - Pruebas de rendimiento
      - Pruebas de carga y estabilidad
      - Pruebas de capacidad o estrés
      - Prueba de escalabilidad
- Pruebas de aceptación.

##### 4.1. **ESPECIFICACIÓN DEL ENTORNO DE PRUEBAS**

Tal y como se ha comentado en el apartado de Análisis funcional, los entornos disponibles para la ejecución de las pruebas son el entorno de **desarrollo** y el entorno de **pre-producción**.

A continuación, se detallan las herramientas que deberán ser utilizadas para la definición, implementación y ejecución de las pruebas definidas en el Plan de Pruebas.

Las versiones de software indicadas a continuación corresponden a las versiones existentes en el momento de definición de esta metodología. Dicho software se irá actualizando con el paso del tiempo por lo que, antes de abordar la implementación del plan de pruebas, los proveedores deberán cerciorarse de las versiones en uso.

##### 4.1.1. **Plataforma de Gestión de Pruebas: TestLink**

La SGSICS dispone del sistema de Gestión de Pruebas **TestLink** 1.9.16.

El Área de Desarrollo proporcionará a sus proveedores acceso a dicho sistema con objeto de que estos procedan a la *carga de los **Planes de Pruebas** definidos y reporte de los resultados de la ejecución de los mismos.*


*Los Planes de Pruebas identificarán y detallarán cada uno de los **Casos de Pruebas** que lo componen, así como los **Requisitos** cubiertos con cada Caso de Prueba.*

*Los casos de prueba deberán ser agrupados por tipología /niveles (funcionales, no funcionales...) y dentro de estos deberán ser agrupados a su vez por grupos funcionales.*

*Por ejemplo:*

**GF Grupos Funcionales**

**GF01 Gestión de Expedientes**

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

*CU001 Dar de alta de Expediente*

*(Aquí se pone el caso de uso y se definen los caminos a recorrer)*

*GNF Grupos No funcionales*

TestLink ofrece mecanismos de **importación/exportación** de Requisitos y Casos de Pruebas, lo cual permite a los proveedores que desarrollen su actividad fuera de las instalaciones del MIR, el alimentar el sistema de la SGSICS a partir de la información definida en su propio sistema.

Por otra parte, el Área de Desarrollo proporcionará las credenciales acceso (\*) a TestLink necesarias para el **reporte automatizado de resultados de la ejecución de las distintas pruebas**.

#### **4.1.2. Framework para Pruebas unitarias: JUnit**

JUnit es el Framework de código abierto (Open Source) elegido por la SGSICS para la automatización de las pruebas unitarias y de integración en proyectos de desarrollo en java.

JUnit facilita el realizar pruebas de regresión de código cuando éste ha sufrido modificaciones.

El framework provee herramientas, clases y métodos que le facilitan la implementación de pruebas de código para asegurar su consistencia y funcionalidad.

#### **4.1.3. Framework de Pruebas automatizadas: Selenium**

En lo relativo a las **pruebas funcionales automatizadas**, la SGSICS establece “Selenium” como framework de pruebas para aplicaciones web.

#### **4.1.4. Herramienta para pruebas de servicios web: SoapUI**

SoapUI 5.1.3 es la herramienta de código abierto elegida por la SGSICS para las pruebas de servicios web.

Esta herramienta permite probar, simular y generar código de servicios web de forma ágil, partiendo del contrato de los mismos en formato WSDL y con vínculo SOAP sobre HTTP

#### **4.1.5. Herramienta para pruebas de rendimiento y escalabilidad: JMeter**

JMeter 3.3 es la herramienta de código abierto elegida por la SGSICS que permite el diseño y ejecución de las pruebas de carga de comportamiento funcional y medición de rendimiento.

#### **4.1.6. Herramientas para pruebas de regresión:**

En lo relativo a las **pruebas funcionales automatizadas**, como ya se indicó en el apartado de framework de Pruebas Automáticas, la SGSICS establece “Selenium” como framework de pruebas para aplicaciones web, se escogerá un subconjunto de pruebas automáticas que han sido creadas para las pruebas automáticas, según el criterio del equipo de desarrollo, del responsable funcional y sobretodo la realización de un análisis sobre el impacto que pudiera tener en el caso de la corrección de una incidencia o un evolutivo.

En lo relativo a las **pruebas de rendimiento y estabilidad**, como ya se indicó en el apartado de herramienta para pruebas de rendimiento y estabilidad, la SGSICS establece “jMeter” como herramienta. Se usarán los scripts anteriormente creados para realizar las pruebas, se decidirán el objetivo de la prueba, teniendo en cuenta los resultados de las pruebas anteriores realizada, para detectar la posible degradación o mejora de la aplicación.

#### 4.1.7. Herramientas para pruebas de API o Contrato:

Actualmente, la SGSICS está abordando un desarrollo con la arquitectura de microservicio, en este contexto tiene en estudio una serie de herramientas para decidir cuál de las herramientas se adapta a sus necesidades.

## 4.2. ESPECIFICACIÓN TÉCNICA DE NIVELES DE PRUEBA

*Para cada uno de los niveles de prueba identificados se deberá proporcionar el diseño detallado de las pruebas que lo componen.*

*A continuación, se enumeran los detalles a incluir en la especificación de las pruebas.*

- *Ámbito de aplicación (prueba unitaria, de integración, del sistema, de implantación o aceptación) y objetivo.*
- *Casos de prueba asociados: se definen en detalle los casos de prueba y se detalla cómo proceder en la ejecución de dichos casos, describiendo todas las entradas necesarias para ejecutar la prueba, y las relaciones de secuencialidad existentes entre las entradas, así como todas aquellas salidas que se espera obtener una vez ejecutado el caso de prueba, y las características especiales requeridas, como, por ejemplo, tiempo de respuesta.*
- *Entorno de prueba: herramientas adicionales, condicionantes especiales de ejecución, etc.*
- *Criterios de aceptación de la prueba.*

*Esta información deberá ser documentada mediante la herramienta **TestLink**.*

*En la herramienta TestLink deberán documentarse también la **Especificación de Requisitos** del sistema identificando los casos de prueba que dan cobertura a estos.*

La **cobertura de casos de prueba de los requisitos funcionales** será en función de la criticidad del requisito funcional. Se tendrán las siguientes tres situaciones

Prioridad Requisito Funcional	Cobertura Casos de Prueba Requisitos Funcionales
Alta	Será necesario identificar y documentar tantos casos de prueba como sean necesarios para que toda la funcionalidad del requisito quede cubierta. Para cubrir toda la casuística de un requisito funcional será necesario crear un caso de prueba por cada uno de los caminos principales, caminos secundarios, caminos de error funcional y caminos de no éxito existentes en la funcionalidad del requisito.
Media	Será necesario identificar tantos casos de prueba como sean necesarios para que toda la funcionalidad del requisito quede cubierta. Solo será necesario documentar un caso de prueba que cubra el camino principal del requisito funcional, quedando como opcional la posibilidad de documentar todos aquellos



	caminos que se consideren importantes.
Baja	Solo será necesario identificar y documentar un caso de prueba que cubra el camino principal del requisito funcional.

Todos los requisitos funcionales (casos de uso) deberán quedar cubiertos por al menos dos casos de prueba (prueba positiva y negativa).

La **cobertura** de las **pruebas funcionales automatizadas** deberá, como mínimo, cubrir los módulos o procesos "críticos" del negocio al que da soporte el sistema, es decir, aquellos procesos que no se pueden detener pues ponen en peligro la continuidad (requisitos de prioridad alta).

#### 4.2.1. Especificación técnica Pruebas de Rendimiento

El **Plan de Pruebas de Rendimiento** se define como un conjunto de casos de prueba que permiten verificar si el sistema cumple con las necesidades de rendimiento que se le exigen en base a los requisitos identificados para el mismo (número de usuarios concurrentes, tiempos de respuesta esperados, escalabilidad del sistema, etc.).


En la fase de diseño se deberán **definir los casos de prueba** incluyendo la descripción, objetivo, prioridad o importancia y declaración de ficheros de datos de entrada que serán necesarios utilizar. Por otro lado, será necesario que se realice su trazabilidad con el requisito de rendimiento que cubra dicho caso.

Una vez terminado el plan, todos los requisitos de rendimiento deberán quedar cubiertos por casos de prueba y viceversa.

#### 4.3. REVISIÓN DE LA PLANIFICACIÓN DE PRUEBAS

*En esta tarea **se completa** y especifica **la planificación de las pruebas**, determinando los distintos perfiles implicados en la preparación y ejecución de las pruebas y en la evaluación de los resultados, así como el tiempo estimado para la realización de cada uno de los niveles de prueba, de acuerdo a la estrategia de integración establecida.*



 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

## 5. **FASE DE CONSTRUCCIÓN Y EJECUCIÓN DE LAS PRUEBAS**

### 5.1.1. **Pruebas unitarias (JUnit)**

Las pruebas unitarias de código han de implementarse a la par que se implementan los diferentes módulos que componen la aplicación.

La cobertura de estas pruebas abarcará, como mínimo, las principales clases de negocio y el 80% del código global.

La compilación y ejecución de las pruebas unitarias serán orquestadas por la plataforma de Integración Continua y se implementarán con total independencia del IDE de desarrollo.

### 5.1.2. **Pruebas funcionales automatizadas (Selenium)**

Ver “22-PRUEBAS/MIR-NORMA-PrFuncAutJava.docx”.

### 5.1.3. **Pruebas funcionales de servicios web (SoapUI)**

Haciendo uso de la herramienta SoapUI el proveedor deberá crear las TestSuites y TestCase necesarias para cubrir las pruebas funcionales de los servicios web implementados. Dichas pruebas serán útiles como pruebas de regresión de los correspondientes servicios web.

Con objeto de que dichas pruebas puedan ser ejecutadas de forma ágil y automatizada el proveedor deberá facilitar los ficheros batch necesarios para la ejecución de las testsuites desde línea de comandos (testrunner.bat) con objeto de que dichas baterías de pruebas puedan ser ejecutadas sin necesidad de abrir la aplicación.

Los casos de prueba (TestCase) deberán manejar juegos de datos de entrada que permitan realizar pruebas funcionales de los servicios web cubriendo tanto los casos de éxito como los de error.


### 5.1.4. **Pruebas de rendimiento (JMeter)**

Ver “22-PRUEBAS/MIR-NORMA-PrRendAut.docx”.

### 5.1.5. **Pruebas de Regresión**

Las pruebas de regresión se basan en las pruebas que anteriormente se han realizado. En el caso de las pruebas de regresión funcionales automáticas se usarían un subconjunto de las pruebas ya desarrolladas y que han aplicado la normativa de pruebas funcionales automáticas java (22-PRUEBAS/MIR-NORMA-PrFuncAutJava.docx), y si fuera necesario bajo el criterio del equipo de desarrollo y el responsable funcional la creación de nuevas pruebas si considera necesarios para la calidad del producto.

Se usará el mismo criterio que se ha indicado del apartado anterior respecto las pruebas de rendimiento y estabilidad, si fuera necesario crear nuevas pruebas se seguiría la normativa (22-PRUEBAS/MIR-NORMA-PrRendAut.docx)

 <b>MINISTERIO DEL INTERIOR</b>	Guía de elaboración planes de pruebas	SECRETARÍA DE ESTADO DE SEGURIDAD
		SUBDIRECCIÓN GENERAL DE SISTEMAS DE INFORMACIÓN Y COMUNICACIONES PARA LA SEGURIDAD

#### **5.1.6. Pruebas de API o Contrato (Solo MicroServicios)**

En los microservicios su parte importante es su comunicación entre los diferentes servicios, entonces la batería de pruebas se debe centrar en sus APIs, para garantizar que la comunicación se realiza correctamente. En estas pruebas se deben incluir la verificación de los esquemas de solicitud y respuesta de la API probada, además de su comportamiento, por ejemplo, que información retorna un endpoint con diferentes solicitudes.

Actualmente, la SGSICS está abordando un desarrollo con la arquitectura de microservicio, por lo que no existe una normativa para este tipo de pruebas, en el futuro se desarrollará una normativa hasta la aparición de esta, se pedirá que se siga el estándar correspondiente que esté definido y buenas prácticas en este campo.

### **6. FASE DE ACEPTACIÓN DEL SISTEMA**

Las pruebas de aceptación tienen como fin validar que el sistema cumple los requisitos básicos de funcionamiento esperado y permitir que el usuario determine la aceptación del sistema. Por este motivo, estas pruebas son realizadas por el usuario final que, durante este periodo de tiempo, debe plantear todas las deficiencias o errores que encuentre antes de dar por aprobado el sistema definitivamente.

El Plan de Aceptación será el documento de referencia que permitirá al usuario validar el cumplimiento de los requisitos acordados como último paso previo a la puesta en explotación.