Guía de desarrollo, Anexo 03.05

Procedimiento de Pruebas Funcionales

Autor: Oficina de Pruebas

GERENCIA INFORMÁTICA
JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 Introducción

1.1 Objetivo

El presente documento establece un procedimiento que detalla los pasos que son necesarios para realizar pruebas funcionales de una aplicación en la DGT.

1.2 Alcance

La normativa que aquí se expone es **de obligado cumplimiento** para todas aquellas personas que vayan a realizar un proceso de pruebas funcionales sobre una aplicación en la DGT: La Oficina de Pruebas, Empresas externas de Desarrollo, etc…

1.3 Glosario

Los términos y acrónimos que se utilizan en este documento y en el resto de documentos de la guía se encuentran recogidos por orden alfabético en el Anexo 30. Glosario con el objetivo de facilitar su lectura y comprensión

2 Procedimiento de Pruebas Funcionales

2.1 Análisis de pruebas

Esta actividad consiste en realizar un análisis de la fase de pruebas funcionales de una aplicación con el fin de conocer el nivel de certificación de la aplicación, así como el nivel de profundidad de la fase de pruebas para las distintas funcionalidades del aplicativo.

2.1.1 Criterios de certificación funcional

Se proporcionan criterios que permitan a los responsables del proyecto DGT establecer de una forma objetiva y normalizada la criticidad de las aplicaciones, con el fin de conocer el nivel de certificación funcional que se desea realizar sobre la misma.

Los criterios son los siguientes:

- Valoración subjetiva: valoración inicial de criticidad del sistema por parte del responsable
- Tipo de producto (Comercial, Desarrollo a medida, Portal…)
- Aplicación web o abierta al exterior (Extranet, Internet)
- Volumen de usuarios al que da/dará servicio
- Frecuencia de uso del sistema
- Entorno tecnológico
- Aplicación centralizada o distribuida
- Información gestionada (datos): Centralizados o Distribuidos
- Funcionalidad crítica/importante para el negocio
- Proporciona información a otras aplicaciones con funcionalidades importantes o críticas
- Valoración del impacto de fallo/no disponibilidad de la aplicación en el negocio
- Posibilidad de alternativas en periodos de fallo
- Impacto en la imagen del organismo en caso de fallo
- Frecuencia de cambios que soporte
- Volumen de información que maneja
- Número de incidencias/fallos detectados con anterioridad

- Confianza en el proveedor de desarrollo
- Tiempos y plazos que se dispone para realizar el desarrollo

2.1.2 Requisitos funcionales

Los **requisitos funcionales** son especificaciones relacionadas con el funcionamiento del sistema, como interactúa con su entorno y cuáles van a ser su estado y funcionamiento, así como indicar lo que el sistema no debe hacer.

Los requisitos deben estar priorizados y dicha priorización debe realizarla preferiblemente el jefe de proyecto DGT, junto con los usuarios.

Una correcta priorización de los requisitos es clave para la fase de pruebas, ya que de esta priorización se obtendrá la profundidad de las pruebas funcionales a realizar.

A continuación se detallan algunos criterios que pueden servir como guía para la priorización de los requisitos:

- Funcionalidad crítica/importante para el negocio
- Volumen de usuarios al que da/dará servicio
- Proporciona información a otras aplicaciones con funcionalidades importantes o críticas
- Valoración del impacto de fallo/no disponibilidad de la aplicación en el negocio
- Posibilidad de alternativas en periodos de fallo
- Impacto en la imagen del organismo en caso de fallo
- Frecuencia de uso del proceso
- Complejidad del proceso
- Solución técnica que lo soporta
- Frecuencia de cambios que soporte
- Volumen de información que maneja
- Número de incidencias/fallos detectados con anterioridad
- Confianza en el proveedor de desarrollo
- Tiempos y plazos que se dispone realizar el desarrollo

Los requisitos funcionales que tengan prioridad superior a 3 se definirán como funcionalidades críticas de la aplicación, y su criticidad se verá reflejada en los planes de prueba funcionales con una mayor cobertura de casos de prueba que los requisitos funcionales de prioridad 3 o inferior.
Para proyectos evolutivos y correctivos se deben identificar los requisitos funcionales que no han sufrido modificaciones con respecto a la versión anterior y que siguen aplicando en la actual, ya que estos requisitos determinarán los casos de prueba de regresión en la fase de especificación de pruebas.

Según el escenario de certificación funcional elegido para una aplicación existen dos situaciones distintas para los requisitos funcionales:

- Escenario 1. Se realizarán pruebas funcionales de la aplicación bajo la metodología de pruebas establecida por la DGT, y serán auditadas y ejecutadas por la oficina de Pruebas.
- Escenario 2. Se realizarán pruebas funcionales de la aplicación bajo la metodología de pruebas establecida por la DGT, serán auditadas por la oficina de pruebas pero no ejecutadas sino que serán ejecutadas por el equipo del proyecto.

2.2 Especificación de pruebas

El plan de pruebas funcional se define como un conjunto de casos de prueba que permiten verificar que el sistema cumple las necesidades establecidas por el usuario en los requisitos.

2.2.1 Cobertura de pruebas

En función de la criticidad de un requisito funcional, la cobertura de casos de prueba que debe crearse será distinta. Se tendrán las siguientes tres situaciones:

- **Requisitos funcionales de prioridad 4 y 5.** Será necesario 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.
- **Requisitos funcionales de prioridad 3.** Será necesario documentar al menos 2 casos de prueba, uno siempre será el que cubra el camino principal del requisito funcional, y el otro

podrá ser un camino secundario, o de error o de no éxito. Queda como opcional la posibilidad de documentar todos aquellos caminos que se consideren importantes.

- **Requisitos funcionales de prioridad 1 o 2.** Solo será necesario documentar un caso de prueba que cubra el camino principal del requisito funcional.

Documentar un caso de prueba es proporcionar información detallada del caso de prueba según las indicaciones del apartado Diseño de los casos de prueba.

Camino Principal. Describe los pasos normales, entendiendo como normal el proceso esperado, que se realizan durante la interacción con el sistema, cuya ejecución culmina con la obtención de los datos esperados. Se podría definir como el escenario exitoso de inicio a fin.

Caminos Secundarios o Alternativos. Son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados. Describen un conjunto de pasos secuenciales originados a partir de la evaluación de una condición o regla, cuyo origen es un paso del camino principal.

Caminos de error funcional. Son los caminos que despliegan los mensajes de error o precaución, generalmente cuando se usan datos o acciones inválidas.

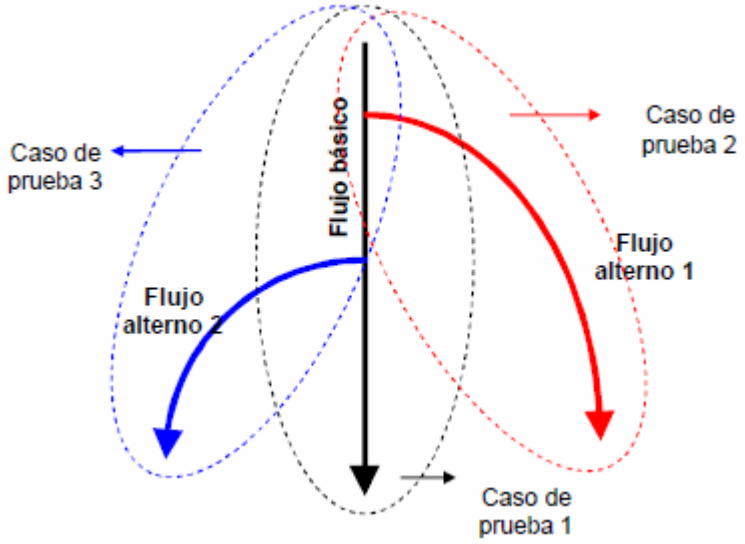


Ilustración 1. Flujos de pruebas

Para proyectos evolutivos y correctivos el plan de pruebas estará formado por:

- Casos de prueba nuevos. Se corresponden con los requisitos nuevos o modificados con respecto a la versión anterior.
- Casos de prueba de regresión. Se corresponden con los requisitos no modificados con respecto a la versión anterior.

2.2.2 Consideraciones para el diseño de los casos de prueba

A continuación se indican una serie de consideraciones a tener en cuenta en el diseño de los casos de prueba:

- **Dos enfoques de pruebas**. Los casos de prueba deben tener dos enfoques, por un lado debe verificar que el software hace lo que se espera de él, y segundo y menos obvio, verificar que el software no hace lo que no se espera de él.
- **Máxima cobertura funcional**. Un caso de prueba debe diseñarse para que cubra el máximo número de posibilidades de entrada diferentes que den como resultado un comportamiento común y único a nivel funcional con el fin de reducir el total de casos. Como posibilidades de entrada se pueden definir conjuntos de datos válidos y/o no válidos.

Ejemplo: Se toma un proceso de negocio típico como es la búsqueda de entidades mediante un grupo de campos:

Proceso de negocio: Búsqueda por los siguientes campos: Código, DNI, apellidos.

Se crearía un caso de prueba cuyo objetivo sería validar el correcto funcionamiento de la función búsqueda donde los campos código, dni, y apellidos estarían parametrizados, y también el resultado esperado (columna “Mensaje mostrado”). En el fichero de datos de entrada se introduciría el juego de datos necesario para cubrir las siguientes pruebas:

	Código	DNI	Apellidos	Mensaje mostrado
Prueba 1	X			Resultado encontrado
Prueba 2		X		Resultado encontrado

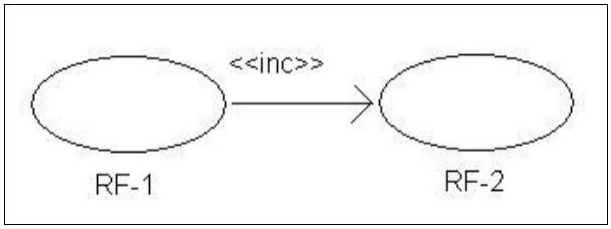
Prueba 3			X	Resultado encontrado
Prueba 4	X	X		Resultado encontrado
Prueba 5	X		X	Resultado encontrado
Prueba 6		X	X	Resultado encontrado
Prueba 7	X	X	X	Resultado encontrado
Prueba 8				Debe introducir al menos un criterio
Prueba 9	Código Inexistente			Resultado no encontrado
Prueba 10		DNI inexistente		Resultado no encontrado
Prueba 11			Apellidos inexistente	Resultado no encontrado

- **Independencia funcional**. Los casos de prueba deben diseñarse lo más independientes posibles desde el punto de vista funcional, es decir, un caso de prueba debe probar caminos funcionales independientes dentro del aplicativo. No deben existir dependencias de casos de prueba por cuestión de datos, es decir, no pueden existir casos de prueba que sean ejecutados previamente como generadores de datos de otros casos de prueba. Para evitar esto el juego de datos tiene que ser lo suficientemente amplio como para probar de forma independiente los diferentes caminos de toda la funcionalidad.

- **Reutilización.** Existen casos de prueba que se corresponden con procesos de negocio que aparecen en repetidas ocasiones en el aplicativo, de manera que es necesario identificarlos dentro del plan de pruebas para que puedan ser reutilizados dentro de otros casos de prueba haciendo el plan de pruebas, reutilizable y mantenible. Siempre teniendo en cuenta la independencia funcional de los casos de prueba.

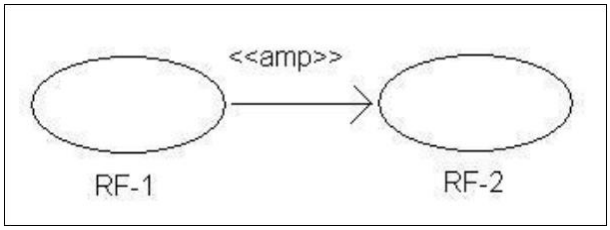
- **Relación entre requisitos funcionales**. En el diseño de los casos de prueba se debe tener en cuenta la relación de los requisitos funcionales entre sí. Se darán dos situaciones:

Relación de inclusión. El RF-1 siempre utiliza la funcionalidad del RF-2.



Todos los casos de prueba diseñados para el RF-1 deben incluir en sus pasos el RF-2.

Relación de ampliación. El RF-1 amplía la funcionalidad del RF-2.



Los casos de prueba se deben diseñar para RF-2 deben contemplar las dos situaciones siguientes:

- Casos de prueba con la funcionalidad de RF-2.
- Casos de prueba con la funcionalidad de RF-2 + RF-1.

- **Parametrización de datos de entrada**. La parametrización se define como la creación de variables en los casos de prueba que alojan el juego de datos, de manera que permite ejecutar el caso de prueba con valores de datos diferentes. Se deben parametrizar todos los datos de entrada del proceso de negocio y proporcionar al menos 3 juegos de datos para ejecutar el caso de prueba. Más información en el "Anexo 03.01 Guia de datos de entrada de pruebas".

- **Parametrización de resultados esperados**. Solo se puede parametrizar los resultados esperados en el caso que el patrón de comportamiento sea el mismo y no implique una ramificación del proceso, ya que esto implicaría otro caso de prueba verificando un camino alternativo. Por ejemplo, un caso permitido sería el alta de un DNI en el que todas las acciones son iguales y el resultado es diferente, por tanto se puede parametrizar el resultado final esperado:

"El DNI 111111111C se ha dado de alta correctamente"

"El DNI 111111111C no se ha dado de alta, ya existe"

2.2.3 Diseño de los casos de prueba

Un **caso de prueba** (GUI y servicios SOAP, REST,…)) se define como un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados, y tiene un objetivo concreto (probar algo). Los casos de prueba deberán ser escritos con el detalle suficiente para que cualquier persona sin ningún conocimiento funcional sobre la aplicación y sin utilizar la intuición, sea capaz de ejecutar las pruebas y encontrar defectos.

Un caso de prueba documentado debe tener la siguiente información:

- Descripción. Descripción sobre el aspecto de la funcionalidad que se está probando.
- Objetivo. Propósito del caso de prueba.
- Condiciones de ejecución. Condiciones que se deben cumplir antes de iniciar el caso de prueba.
- Prioridad. Se definen cinco niveles de prioridad para los casos de prueba: 1-Low, 2-Medium, 3-High, 4-Very High y 5-Urgent.
- Datos de entrada. Lista de variables y valores usados para la ejecución de la prueba. Estos valores se escriben por línea de manera que cada línea se corresponde con una ejecución del caso de prueba.
- Diseño de pasos. Es la secuencia exacta de acciones necesarias para completar el caso de prueba. Cada paso debe tener la siguiente información:
    - Descripción. Son las acciones de usuario que se deben realizar sobre el aplicativo.
    - Resultado esperado. Son las verificaciones que se deben realizar tras la ejecución del paso para comprobar que el paso del proceso de negocio se ha realizado correctamente.
    - Captura imagen. Se anexarán las capturas de pantalla necesarias que aporten un valor extra para poder ejecutar correctamente el caso de prueba.

- Adjuntos: en caso necesario se puede adjuntar a cada caso de prueba los ficheros necesarios, por ejemplo, para el caso de servicios web el xml/json de petición y respuesta correspondientes a modo de ejemplo. En algunos casos donde la parametrización de los datos de los servicios web sea complicada (por ejemplo, en caso de servicios web firmados), se pueden adjuntar al propio caso los XML/JSON a usar como datos, en ese caso habrá que proporcionar 3 ficheros de entrada distintos como sucede en el caso de que los datos estén en el XLS.

Una vez definidos los casos de prueba para un requisito, se debe asignar la prioridad de cada uno de ellos dentro del contexto funcional al cual pertenece, ya que esta priorización es utilizada para identificar como de importante (probabilidad de encontrar defectos en la aplicación) es el caso de prueba dentro del contexto en el cual está definido.

A continuación se dan algunas recomendaciones que pueden ayudar a definir la prioridad de los casos de prueba:

- Importancia funcional del camino que ejecuta. Dentro de todos los caminos posibles de un requisito funcional deberá existir una escala de importancia según la funcionalidad que ejecutan. Al menos para los requisitos funcionales de prioridad 4 o 5, los casos de prueba de los caminos principales deberán tener prioridad 4 o 5.
- Revisar los resultados de ejecuciones anteriores. Depende de si se han encontrado un gran número de defectos en ejecuciones anteriores o si simplemente se ha añadido nueva funcionalidad.
- Identificar en qué áreas los defectos han sido arreglados. Si existen áreas con muchos defectos arreglados esto significa que nuevo código ha sido añadido, quitado o modificado, por lo que implica que los casos de prueba afectados por estas modificaciones deben ser ejecutados y se les debe asignar una prioridad mayor.
- Identificar las nuevas funcionalidades. Los casos de prueba que hayan sufrido modificaciones por añadirse nueva funcionalidad o cambios en requisitos deben de ser identificados y ser asignados con prioridad mayor.

Cada caso de prueba deberá estar trazado con uno o varios requisitos funcionales según la funcionalidad que verifique.

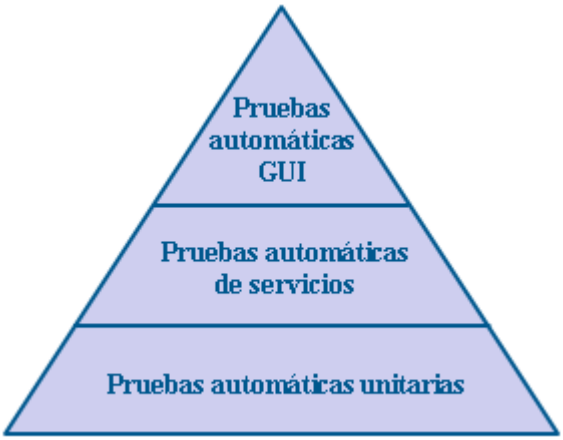
2.3 Automatización de pruebas funcionales

Para aplicaciones que ya hayan superado su primera release y que sean estables se puede realizar la automatización de su plan de pruebas funcional de modo que se comprueben los procesos de negocio más críticos de la aplicación de acuerdo a los criterios que se exponen a continuación. El objetivo es disponer de una serie de casos de prueba funcionales automatizados que permitan conocer de una forma rápida y fiable el estado de la aplicación.

La automatización de los casos de prueba se realizará por parte de la Oficina de Pruebas a partir del conjunto de casos de prueba funcionales que cumplan los requisitos de automatización:

- Estabilidad de la aplicación: la aplicación a automatizar debe ser estable
- Juego de datos reutilizable: los datos necesarios para la ejecución de las pruebas debe poder ser reutilizados en posteriores ejecuciones (no se “quema” el dato)
- Con resultados predecibles: la aplicación debe comportarse siempre del mismo modo para un conjunto de datos de entrada y características del entorno iguales
- Pruebas repetitivas: las pruebas a automatizar deben poder realizarse de manera habitual, no siendo pruebas de una única ejecución.

Se automatizarán tanto casos de prueba GUI como API, dando prioridad a los test de APIs (ya que es donde reside el core de la aplicación) siguiendo las tendencias de automatización actuales tal y como queda reflejado en la pirámide de Cohn.



La información necesaria para la automatización será el propio caso de prueba funcional perfectamente definido como se indica en el punto “2.2.3 Diseño de los casos de prueba”, con el juego de datos aplicable, descripción funcional de los pasos de ejecución y de los resultados esperados.

Los casos de prueba a automatizar serán como normal general:

- Todos los casos de prioridad 4 y 5
- Los casos de prioridad 3 de requisitos de prioridad 4 y 5

Excepciones:

- Para webservices se automatizarán todos los casos independientemente de su prioridad si es viable
- Para aplicaciones con un bajo número de casos en el Plan de Pruebas se automatizarán todos si es viable

En caso de ser posible, las pruebas funcionales automatizadas también se podrían usar en el entorno de desarrollo para ello se debería proporcionar un juego de datos completo en dicho entorno.

.

2.4 Ejecución de pruebas

En esta fase se realiza la ejecución de tres tipos de pruebas, pruebas manuales, pruebas de regresión automatizadas y pruebas exploratorias.

2.4.1 Baterías de pruebas

2.4.1.1Pruebas manuales

En esta fase se diseñan las baterías de pruebas funcionales, que son un conjunto de casos de prueba que se ejecutan a la vez y que tienen un objetivo común.

La selección de los casos de prueba para la formación de las baterías de prueba se realizará mediante la priorización asignada a los requisitos funcionales y a los casos de prueba, de manera que serán ejecutados primero los de prioridad más alta y se irá bajando en profundidad. Se ejecutarán primero todos los casos de prueba de prioridad 4 y 5.

No obstante, el jefe de proyecto DGT de la aplicación puede dirigir las baterías de ejecución de pruebas hacia aquellas funcionalidades de la aplicación que por una situación concreta necesiten pruebas más exhaustivas. Por ejemplo, probar áreas funcionales de la aplicación donde muchos defectos han sido arreglados, lo que significa que nuevo código ha sido añadido, modificado o eliminado; probar áreas funcionales nuevas, o probar áreas que no han sido modificadas pero se ven directamente afectadas por las nuevas.

Cada ciclo de pruebas debe estar asociado a una revisión de la aplicación desplegada en el entorno de preproducción, y debe tener asociado un juego de datos actualizado a esa nueva versión (consultar el Anexo 03.01 Guía de datos de entrada de pruebas).

Durante el transcurso de la ejecución de pruebas se actualizará el estado de los casos de prueba de manera que se permite conocer en todo momento el grado de avance de las pruebas sobre el sistema.

En la herramienta de gestión del ciclo de pruebas está disponible toda la información relativa a la ejecución de los casos de prueba (procedimiento, resultado esperado y resultado obtenido) para su revisión por Desarrollo una vez finalizada la ejecución.

2.4.1.2Pruebas de regresión funcionales automatizadas

Si se dispone de casos de prueba automatizados se ejecutarán periódicamente con el objetivo de comprobar que la aplicación funciona correctamente en todo momento y detectar precozmente errores en nuevas versiones (en aquellos casos que no haya habido cambio de funcionalidad). En caso de modificaciones será necesario adaptar el Plan de Pruebas Funcional por parte de Desarrollo para posteriormente adecuar los scripts automáticos en consonancia.

2.4.1.3Pruebas exploratorias

Durante la ejecución de las pruebas funcionales manuales documentadas la oficina de Pruebas ejecutará de forma paralela una batería de pruebas funcionales exploratorias o aleatorias.

Las **pruebas exploratorias** son pruebas no sistemáticas que se basan en la experiencia, conocimiento e intuición de los probadores para encontrar errores. En estas pruebas no existe un guion predefinido, de manera que el probador es libre de tomar cualquier camino alternativo no calculado o no premeditado. Son siempre pruebas complementarias a las pruebas sistemáticas.

Como orientación se enumeran algunas condiciones genéricas que se aplican a las pruebas exploratorias de cualquier aplicación:

- Probar toda la navegación del sitio (Links rotos, backspace, forward button, etc)
- Cambiar las resoluciones del monitor y comprobar cómo se ven los elementos.
- Probar con varios navegadores
- Probar el movimiento del cursor entre los diferentes campos
- Introducir datos con caracteres especiales o no permitidos, números negativos, comas, paréntesis, comillas, etc.
- Introducir texto con contenido HTML o muy grande
- Cargar páginas que necesitan autenticación directamente en el navegador sin estar logado
- Probar con diferentes sistemas operativos
- Probar diferentes formatos de fecha

- Probar la correcta paginación
- …….

Es importante destacar el ámbito funcional concreto de cada aplicación en la elaboración de las pruebas exploratorias y que debe basarse en el conocimiento funcional adquirido en las fases anteriores de prueba y en la experiencia del tester.