

## Tipos de prueba

Un tipo de prueba es un grupo de actividades de pruebas destinadas a probar las características específicas de un sistema de software, o de una parte de un sistema, basados en objetivos de pruebas específicas.

Dichos objetivos pueden incluir:

1. Evaluar las características de calidad funcional tales como la completitud, corrección y pertinencia.
2. Evaluar características no funcionales de calidad, tales como la fiabilidad, eficiencia de desempeño, seguridad, confiabilidad y usabilidad.
3. Evaluar si la estructura o arquitectura del componente o sistema es correcta, completa y según lo especificado.
4. Evaluar los efectos de los cambios, tales como confirmar que los defectos han sido corregidos (prueba de confirmación) y buscar cambios no deseados en el comportamiento que resulten de los cambios en el software o en el entorno (prueba de regresión)

	1. Prueba Funcional	2. Prueba No Funcional	3. Prueba Estructurales	4. Prueba Asociada al Cambio
<b>Definición</b>	La prueba funcional de un sistema incluye pruebas que evalúan las funciones que el sistema debe realizar. Las funciones describen <b>qué hace</b> el sistema.	La prueba no funcional prueba " <b>cómo de bien</b> " se comporta el sistema.	Éstas pruebas están basadas en la estructura interna del sistema o en su implementación. La estructura interna puede incluir código, arquitectura, flujos de trabajo y/o flujos de datos dentro del sistema	Existen 2 tipos de prueba relacionadas al cambio: <ul style="list-style-type: none"> <li>• <b>Prueba de confirmación:</b> Una vez corregido un defecto, el software se puede probar con todos los casos de prueba que fallaron debido al defecto, que se deben volver a ejecutar en la nueva versión de software. El objetivo de una prueba de confirmación es confirmar que el defecto original se ha solucionado de forma satisfactoria.</li> <li>• <b>Prueba de regresión:</b> Es posible que un cambio hecho en una parte del código, ya sea una corrección u otro tipo de cambio, pueda afectar accidentalmente el comportamiento de otras partes del código, ya sea</li> </ul>

				dentro del mismo componente, en otros componentes del mismo sistema, o incluso en otros sistemas. La prueba de regresión implica la realización de pruebas para detectar estos efectos secundarios no deseados.
<b>Implementación</b>	La prueba funcional observa el comportamiento del software.	El diseño y ejecución de la prueba no funcional puede implicar competencias y conocimientos especiales, como el conocimiento de las debilidades inherentes a un diseño o tecnología -por ejemplo: vulnerabilidades de seguridad asociadas con determinados lenguajes de programación-.	El diseño y la ejecución de este tipo de pruebas pueden implicar competencias o conocimientos especiales, como la forma en que se construye el código, cómo se almacenan los datos, y cómo utilizar las herramientas de cobertura e interpretar correctamente sus resultados.	Especialmente en los ciclos de vida de desarrollo iterativos e incrementales (por ejemplo, Agile), las nuevas características, los cambios en las características existentes y la refactorización del código dan como resultado cambios frecuentes en el código, lo que también requiere pruebas asociadas al cambio.

<b>Niveles de Prueba</b>	Se pueden realizar pruebas funcionales en todos los niveles de prueba.	Se pueden realizar pruebas no funcionales en todos los niveles de prueba	Se puede realizar en el nivel de componente y de integración.	La prueba de confirmación y la prueba de regresión se realizan en todos los niveles de prueba
<b>Alcance</b>	Los requisitos funcionales pueden estar detallados en los siguientes documentos: especificaciones de requisitos del negocio, épicas, historias de usuarios, casos de uso y/o especificaciones funcionales.	La prueba no funcional del sistema evalúa características como la usabilidad, la eficiencia del desempeño o la seguridad.	En el nivel de prueba de integración de componentes, la prueba estructural pueden basarse en la arquitectura del sistema, como las interfaces entre componentes	
<b>Cobertura</b>	La cobertura funcional es la medida en que algún tipo de elemento funcional ha sido practicado por pruebas, y se expresa como un porcentaje del tipo o tipos de elementos cubiertos.	La cobertura no funcional es la medida en que algún tipo de elemento no funcional ha sido practicado por pruebas, y se expresa como un porcentaje del tipo o tipos de elementos cubiertos.	La cobertura estructural es la medida en que algún tipo de elemento estructural ha sido practicado mediante pruebas, y se expresa como un porcentaje del tipo de elemento cubierto.	Los juegos de prueba de regresión se ejecutan muchas veces y generalmente evolucionan lentamente, por lo que la prueba de regresión es un fuerte candidato para la automatización. La cobertura crece a medida que se agregan más funcionalidades al sistema por lo tanto más pruebas de regresión

## Ejemplos de tipos de prueba

Los siguientes ejemplos están basados en una aplicación bancaria.

### **Pruebas funcionales:**

- Prueba de componente: las pruebas se diseñan con base en la forma en que un componente debe calcular el interés a pagar por un préstamo.
- Prueba de integración de componentes: las pruebas se diseñan en función de cómo la información de la cuenta capturada en la interfaz de usuario se transfiere a la lógica de negocio.
- Prueba de sistema: las pruebas se diseñan sobre la base de cómo los titulares de cuentas pueden solicitar una línea de crédito sobre sus cuentas corrientes.
- Prueba de integración de sistemas: las pruebas se diseñan en función de cómo el sistema utiliza un microservicio externo para comprobar la calificación crediticia del titular de una cuenta.
- Prueba de aceptación: las pruebas se diseñan con base en la forma en que el empleado del banco tramita la aprobación o rechazo de una solicitud de crédito.

### **Pruebas no funcionales:**

- Prueba de componente: las pruebas de rendimiento están diseñadas para evaluar el número de ciclos de CPU necesarios para realizar un cálculo de intereses totales complejo.
- Prueba de integración de componentes: las pruebas de seguridad están diseñadas para vulnerabilidades de desbordamiento de memoria intermedia debido a que los datos pasan de la interfaz de usuario a la lógica de negocio.
- Prueba de sistema: las pruebas de portabilidad están diseñadas para comprobar si la capa de presentación funciona en todos los navegadores y dispositivos móviles soportados.