

# Niveles de pruebas

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree



Los **niveles de prueba** se caracterizan por los siguientes atributos:

- Objetivos específicos
- Bases de prueba
- Objeto de prueba
- Defectos y fallos característicos
- Enfoques y responsabilidades específicas



# Índice

1. [Prueba de componente](#)
2. [Prueba de integración](#)
3. [Prueba de sistema](#)
4. [Prueba de aceptación](#)

1

# Prueba de componente

# Objetivos específicos

:

- Reducir el riesgo.
- Verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados.
- Generar confianza en la calidad del componente.
- Encontrar defectos en el componente.
- Prevenir la propagación de defectos a niveles de prueba superiores.

# Base de prueba

:

Algunos ejemplos de productos de trabajo que se pueden utilizar como base de prueba incluyen:

- Diseño detallado.
- Código.
- Modelo de datos.
- Especificaciones de los componentes.

# Objetos de prueba

:

Los objetos de prueba característicos para la prueba de componente incluyen:

- Componentes, unidades o módulos.
- Código y estructuras de datos.
- Clases.
- Módulos de base de datos.

# Defectos y fallos característicos

:Ejemplos de defectos y fallos característicos de la prueba de componente incluyen:

- Funcionamiento incorrecto —por ejemplo, no lo hace de la manera en que se describe en las especificaciones de diseño—.
- Problemas de flujo de datos.
- Código y lógica incorrectos.



# Enfoque y responsabilidades específicas

En general, el desarrollador que escribió el código realiza la prueba de componente. Los desarrolladores pueden alternar el desarrollo de componentes con la búsqueda y corrección de defectos.

A menudo, estos escriben y ejecutan pruebas después de haber escrito el código de un componente. Sin embargo, especialmente en el desarrollo ágil, la redacción de casos de prueba de componente automatizados puede preceder a la redacción del código de la aplicación.

# 2 | Prueba de integración

# Objetivos

La prueba de integración se centra en las interacciones entre componentes o sistemas.

- Reducir el riesgo.
- Verificar que los comportamientos funcionales y no funcionales de las interfaces sean los diseñados y especificados.
- Generar confianza en la calidad de las interfaces.
- Encontrar defectos —que pueden estar en las propias interfaces o dentro de los componentes o sistemas—.
- Prevenir la propagación de defectos a niveles de prueba superiores.

# Base de prueba

Algunos ejemplos de productos de trabajo que pueden utilizarse como base de prueba para la prueba de integración incluyen:

- Diseño de software y sistemas.
- Diagramas de secuencia.
- Especificaciones de interfaz y protocolos de comunicación.
- Casos de uso.
- Arquitectura a nivel de componente o de sistema.
- Flujos de trabajo.
- Definiciones de interfaces externas

# Objetos de prueba

Los objetos de prueba característicos para la prueba de integración incluyen:

- Subsistemas.
- Bases de datos.
- Infraestructura.
- Interfaces.
- Interfaces de programación de aplicaciones —API por sus siglas en inglés—.
- Microservicios.

# Defectos y fallos característicos

- Datos incorrectos, datos faltantes o codificación incorrecta de datos.
- Secuenciación o sincronización incorrecta de las llamadas a la interfaz.
- Incompatibilidad de la interfaz.
- Fallos en la comunicación entre componentes.
- Fallos de comunicación entre componentes no tratados o tratados de forma incorrecta.
- Suposiciones incorrectas sobre el significado, las unidades o las fronteras de los datos que se transmiten entre componentes.

# Enfoque y responsabilidades específicas

La prueba de integración debe concentrarse en la integración propiamente dicha. Se puede utilizar los tipos de prueba funcional, no funcional y estructural. En general es responsabilidad de los testers.

# 3 | Prueba de sistema



# Objetivos

- Reducir el riesgo.
- Verificar que los comportamientos funcionales y no funcionales del sistema son los diseñados y especificados.
- Validar que el sistema está completo y que funcionará como se espera.
- Generar confianza en la calidad del sistema considerado como un todo.
- Encontrar defectos.
- Prevenir la propagación de defectos a niveles de prueba superiores o a producción.

# Base de prueba

Algunos ejemplos de productos de trabajo que se pueden utilizar como base de prueba incluyen:

- Especificaciones de requisitos del sistema y del software —funcionales y no funcionales—.
- Informes de análisis de riesgo.
- Casos de uso.
- Épicas e historias de usuario.
- Modelos de comportamiento del sistema.
- Diagramas de estado.
- Manuales del sistema y del usuario.

# Objetos de prueba

- Aplicaciones.
- Sistemas hardware/software.
- Sistemas operativos.
- Sistema sujeto a prueba (SSP).
- Configuración del sistema y datos de configuración.

# Defectos y fallos característicos

- Cálculos incorrectos.
- Comportamiento funcional o no funcional del sistema incorrecto o inesperado.
- Control y/o flujos de datos incorrectos dentro del sistema.
- Incapacidad para llevar a cabo, de forma adecuada y completa, las tareas funcionales extremo a extremo.
- Fallo del sistema para operar correctamente en el/los entorno/s de producción.
- Fallo del sistema para funcionar como se describe en los manuales del sistema y de usuario.

# Enfoque y responsabilidades específicas

La prueba de sistema debe centrarse en el comportamiento global y extremo a extremo del sistema en su conjunto, tanto funcional como no funcional. Debe utilizar las técnicas más apropiadas para los aspectos del sistema que serán probados. Los probadores independientes, en general, llevan a cabo la prueba de sistema.

# 4 | Prueba de aceptación

# Objetivos

La prueba de aceptación, al igual que la prueba de sistema, se centra normalmente en el comportamiento y las capacidades de todo un sistema o producto. Los objetivos de la prueba de aceptación incluyen:

- Establecer confianza en la calidad del sistema en su conjunto.
- Validar que el sistema está completo y que funcionará como se espera.
- Verificar que los comportamientos funcionales y no funcionales del sistema sean los especificados.

# Base de prueba

Entre los ejemplos de productos de trabajo que se pueden utilizar como base de prueba se encuentran:

- Procesos de negocio.
- Requisitos de usuario o de negocio.
- Normativas, contratos legales y estándares.
- Casos de uso.
- Requisitos de sistema.
- Documentación del sistema o del usuario.
- Procedimientos de instalación.
- Informes de análisis de riesgo.



# Objetos de prueba

- Sistema sujeto a prueba.
- Configuración del sistema y datos de configuración.
- Procesos de negocio para un sistema totalmente integrado.
- Sistemas de recuperación y sitios críticos —para pruebas de continuidad del negocio y recuperación de desastres—.
- Procesos operativos y de mantenimiento.
- Formularios.
- Informes.
- Datos de producción existentes y transformados.

# Defectos y fallos característicos

Entre los ejemplos de defectos característicos de cualquier forma de prueba de aceptación se encuentran:

- Los flujos de trabajo del sistema no cumplen con los requisitos de negocio o de usuario.
- Las reglas de negocio no se implementan de forma correcta.
- El sistema no satisface los requisitos contractuales o reglamentarios.
- Fallos no funcionales tales como vulnerabilidades de seguridad, eficiencia de rendimiento inadecuada bajo cargas elevadas o funcionamiento inadecuado en una plataforma soportada.

# Enfoque y responsabilidades específicas

A menudo es responsabilidad de los clientes, usuarios de negocio, propietarios de producto u operadores de un sistema, y otros implicados también pueden estar involucrados. La prueba de aceptación se considera, a menudo, como el último nivel de prueba en un ciclo de vida de desarrollo secuencial.

DigitalHouse>  
Coding School