


# Introducción a Quality Assurance

## ¿Qué es un caso de prueba?

Los casos de prueba (Test cases en inglés) definen cómo probar un sistema, software o una aplicación. Un caso de prueba es un conjunto singular de acciones o instrucciones que debe realizar un tester que valida un aspecto específico de la funcionalidad de un producto o aplicación. Si la prueba falla, el resultado podría ser un defecto de software que la organización puede clasificar para arreglar.


Un tester o profesional de control de calidad generalmente escribe casos de prueba que se ejecutan después de completar una característica o el grupo de características que componen la versión. Los casos de prueba también confirman si el producto cumple con los requisitos de software.

Un grupo de casos de prueba se organiza en un conjunto de pruebas, que prueba un segmento lógico de la aplicación, como una característica específica.

**Te invitamos a ver este breve video para completar la definición de Casos de prueba:**  [¿Qué es un Caso de Prueba de Software?](#)

## ¿Qué es un escenario de prueba?

Un escenario de prueba describe una situación o funcionalidad que requiere pruebas. Por ejemplo, un escenario de prueba podría ser: "Verificar la funcionalidad de inicio de sesión". Los escenarios de prueba suelen tener sus propios números de identificación para el seguimiento. Los equipos de control de calidad a menudo derivan casos de prueba (acciones de bajo nivel) a partir de escenarios de prueba (acciones de alto nivel); y los escenarios de prueba generalmente provienen de la documentación de requisitos comerciales y de software.

**Te invitamos a ver este breve video para completar la definición de ESCENARIOS de PRUEBA:**  [¿Qué son los ESCENARIOS de PRUEBA?](#)

## ¿Por qué son importantes los casos de prueba?

Los casos de prueba definen lo que se debe hacer para probar un sistema, incluidos los pasos ejecutados en el sistema, los valores de datos de entrada que se ingresan en el sistema y los resultados que se esperan durante la ejecución del caso de prueba. El uso de casos de prueba permite a los desarrolladores y testers descubrir errores que pueden haber ocurrido durante el desarrollo o defectos que se pasaron por alto durante las pruebas ad hoc<sup>1</sup>.

Los beneficios de un caso de prueba efectivo incluyen:

- Buena cobertura de prueba garantizada.
- Costos reducidos de mantenimiento y soporte de software.
- Casos de prueba reutilizables.
- Confirmación de que el software satisfaga los requisitos del usuario final.
- Mejora de la calidad del software y la experiencia del usuario.
- Los productos de mayor calidad conducen a clientes más satisfechos.
- Más clientes satisfechos aumentarán las ganancias de la empresa.

En general, escribir y usar casos de prueba conducirá a la optimización del negocio. Los clientes están más satisfechos, aumenta la retención de clientes, disminuyen los costos de servicio al cliente y reparación de productos, y se producen productos más confiables, lo que mejora la reputación y la imagen de marca de la empresa.

---

<sup>1</sup> Ad hoc es un término del latín que significa literalmente 'para esto'. Se emplea como locución adjetiva con el sentido de 'adecuado, apropiado, dispuesto especialmente para un fin'; y como locución adverbial, con el sentido de 'a propósito para la ocasión'.

## Tipos de casos de prueba

Para validar y verificar la funcionalidad del sistema, la organización debe adoptar un enfoque multifacético que evalúe la parte delantera y trasera del producto. Hay diferentes formas de categorizar los distintos tipos de casos de prueba.

- **Casos de prueba formales:** Con estos tipos de casos de prueba, el tester escribe una prueba en la que todas las entradas son conocidas y detalladas, como las condiciones previas y los datos de prueba. Las pruebas formales tienen una entrada predefinida, lo que significa que proporcionan una salida esperada, que la prueba intenta validar.
- **Casos de prueba informales:** Estos no tienen entradas o salidas conocidas. Los testers ejecutan este tipo de casos de prueba para descubrir y registrar los resultados, lo que puede revelar hallazgos interesantes sobre la calidad digital.

La mayoría de los tipos de casos de prueba son formales, planificados con anticipación de acuerdo con los requisitos del software. exploremos algunos tipos y ejemplos de casos de prueba más:



**1. Casos de prueba de funcionalidad:** Estas pruebas determinan si la funcionalidad de destino tiene éxito o no al realizar su función dentro del sistema. El equipo de control de calidad escribe estos tipos de casos de prueba en función de los requisitos y los ejecuta cuando el equipo de desarrollo termina con la función. Muchos tipos diferentes de pruebas funcionales pueden validar la funcionalidad de la aplicación, incluidas las pruebas unitarias que verifican los segmentos de funcionalidad aislados más pequeños posibles. Los casos de prueba funcionales deben incluir:

- una descripción y/o nombre de la función bajo prueba
- condiciones previas
- pasos para la prueba
- un resultado esperado

Ejemplo: Realice un inicio de sesión exitoso y valide que el usuario haya iniciado sesión.

**2. Casos de prueba de IU:** Estas pruebas confirman que la interfaz de usuario (con lo que interactúa el usuario final) funciona como se esperaba. Por lo general, las pruebas de IU se enfocan en los elementos visuales de una aplicación o página web para confirmar que funcionan y se desempeñan de acuerdo con los requisitos. Las pruebas de IU a menudo examinan elementos de visualización como menús, submenús, botones, tablas y columnas para asegurarse de que sean legibles y consistentes. Las interfaces de usuario siguen evolucionando. Por esta razón, las pruebas de IU también pueden significar validar una interfaz de voz o video. Las pruebas de IU también deben incluir problemas de accesibilidad, como si un lector de pantalla puede identificar un botón en una página.

Ejemplo: Navegue a la página de inicio, valide que el menú de hamburguesas se muestre correctamente para escritorio y web móvil.

3. **Casos de prueba de rendimiento:** Las pruebas funcionales comprueban si la aplicación funciona. Las pruebas no funcionales, como las pruebas de rendimiento, comprueban el rendimiento de la aplicación en diferentes tipos de cargas de trabajo. Una prueba de rendimiento debe ser específica con cada paso y resultado esperado documentados, así como datos de entrada claramente definidos, para que el tester pueda evaluar con precisión cómo funciona el sistema en las condiciones dadas. Hay una variedad de tipos de pruebas de rendimiento, incluidas pruebas de carga, tensión, picos y escalabilidad. Cada tipo de prueba de rendimiento, y cada prueba individual, revela información diferente sobre cómo responde el sistema a las distintas cargas de usuarios.

Ejemplo: Mida la mayor cantidad de usuarios que un sistema puede manejar antes de que se bloquee.

4. **Casos de prueba de integración:** Estos tipos de casos de prueba evalúan cómo funciona la funcionalidad combinada cuando se fusiona con la aplicación. Si bien es importante probar unidades individuales de software, es igualmente importante asegurarse de que los sistemas dispares puedan comunicarse entre sí de manera efectiva. El tester debe comprender bien los flujos de la aplicación para escribir pruebas de integración efectivas. Las pruebas de API son un aspecto de las pruebas de integración. Las aplicaciones se comunican entre sí a través de las API, especialmente a medida que los productos se interconectan más en el mundo centrado en dispositivos móviles de hoy. Las pruebas de API son un ejercicio vital para cubrir con casos de prueba de integración.

Ejemplo: inicie sesión a través del mercado de un vendedor, valide que el mercado reconozca al usuario como conectado; en otras palabras, los módulos de inicio de sesión y del mercado se comunican entre sí.

5. **Casos de prueba de usabilidad:** En lugar de probar la funcionalidad o el rendimiento de la aplicación, las pruebas de usabilidad examinan lo que los

posibles usuarios finales, no los testers, piensan de un producto. Los investigadores de UX preparan pruebas para participantes fuera de la organización para medir qué tan fácil o difícil es usar el producto. Las organizaciones pueden realizar pruebas de usabilidad de varias maneras, incluidas moderadas o no moderadas, remotas o en persona. El objetivo es aprovechar la perspectiva de un usuario final para identificar puntos en la aplicación que harían que dejara de usarla. Las pruebas de usabilidad pueden ser formales o informales, según el objetivo y el método UX de investigación.

Ejemplo: encargue al participante una transferencia de dinero entre sus cuentas corrientes y de ahorros, luego evalúe si puede completar con éxito la tarea y si experimenta alguna dificultad con el proceso.

6. **Casos de prueba de bases de datos:** El hecho de que la funcionalidad de una aplicación, la interfaz de usuario y las API funcionen no significa que los datos se almacenen correctamente. Las pruebas de la base de datos validan si los datos de la aplicación se almacenan de acuerdo con los requisitos y las reglamentaciones. Al igual que las pruebas de funcionalidad, las pruebas de bases de datos pueden variar en alcance, desde la validación de un pequeño objeto de base de datos hasta una acción compleja que involucra múltiples partes de la aplicación. Algunos criterios que las pruebas de la base de datos pueden evaluar incluyen si los datos se almacenan de manera consistente, si personas no autorizadas pueden acceder a ellos y cómo se almacenan localmente en un dispositivo. Los datos consistentes y seguros deben ser una prioridad para todas las empresas, independientemente de los estándares de cumplimiento de la industria; las pruebas de bases de datos ayudan a lograrlo.

Ejemplo: Valide que los nuevos datos de PII del cliente se almacenen en un formato cifrado.

7. **Casos de prueba de seguridad:** Estas pruebas identifican vulnerabilidades dentro de un sistema o producto. Otro tipo de prueba no funcional, las pruebas de seguridad tienen como objetivo encontrar formas de proteger mejor los activos de software, así como identificar cómo el sistema resiste los tipos comunes de ataques y definir el riesgo asociado con el producto. Algunas pruebas de seguridad pueden incluir análisis de vulnerabilidades, análisis de configuración y pruebas de penetración, también llamadas pruebas intrusivas. En última instancia, el objetivo de las pruebas de seguridad es generar comentarios procesables que la organización pueda usar para remediar las vulnerabilidades.

Ejemplo: valide que no puede acceder a los documentos de la empresa sin un inicio de sesión exitoso.

8. **Casos de prueba de aceptación del usuario:** Estos tipos de casos de prueba validan el producto desde la perspectiva del usuario final. Un usuario final o cliente realiza pruebas de aceptación del usuario en un entorno de prueba para validar el flujo de extremo a extremo del producto. Las pruebas de aceptación del usuario pueden ser útiles cuando los requisitos comerciales cambian durante el curso del desarrollo. Las partes interesadas no siempre comunican de manera efectiva estos cambios al equipo de desarrollo. A través de los casos de prueba UAT, la organización puede documentar los criterios de entrada y salida que cubren las lagunas en las pruebas anteriores.

Ejemplo: validar que un usuario puede registrarse para obtener una nueva cuenta y que recibe una confirmación por correo electrónico.

9. **Casos de prueba exploratorios:** Estos casos de prueba informales ocurren cuando el tester evalúa el sistema de forma ad-hoc para intentar descubrir defectos que no se detectaron en las pruebas estructuradas. Si bien las pruebas exploratorias no están definidas por un conjunto prescrito de acciones, el enfoque aún requiere cierta estructura, particularmente en lo que

respecta a la documentación de resultados y el calendario, para garantizar una retroalimentación efectiva. Las pruebas exploratorias pueden ayudar a validar los requisitos al verificar el sistema de maneras no cubiertas en las pruebas con guión. Las pruebas exploratorias permiten que la organización de control de calidad sea adaptable y aprenda de las lagunas en la cobertura de las pruebas.

Ejemplo: verifique cómo el uso del botón Atrás del navegador afecta la funcionalidad de la aplicación y si requiere otro inicio de sesión.

## Resultados del caso de prueba

Si bien los objetivos de los casos de prueba varían, la mayoría de los casos formales tienen resultados predecibles. De hecho, el formato de caso de prueba típico debe detallar el resultado esperado y el resultado real, que la prueba misma valida. La mayoría de los resultados de los casos de prueba entran en estas categorías:

- aprobado
- fallo
- sin ejecutar
- obstruido

Las pruebas de **aprobación** y **falla** indican que el sistema logra lo que se supone que debe hacer o falla en ese intento. Estos resultados no deben confundirse con las pruebas diseñadas para ser positivas o negativas, que pueden pasar o fallar. Las pruebas positivas aseguran que los usuarios puedan seguir todos los pasos y pasar el resultado esperado cuando la entrada es correcta, como una transferencia de dinero exitosa entre cuentas cuando hay un saldo superior a \$0. Las pruebas negativas aseguran que el sistema maneja correctamente las entradas no válidas, como no permitir el inicio de sesión si la contraseña es incorrecta. Ambos tipos de pruebas pasan o fallan dependiendo del resultado esperado.

Los resultados de las pruebas que se marcan como **no ejecutados** son, como sugiere el nombre, pruebas que aún no se han ejecutado o que no se ejecutarán como parte de esta ronda de pruebas. Las pruebas bloqueadas resultan de una



circunstancia externa o condición previa que inhibe la ejecución de la prueba. Por ejemplo, una falla del sistema que impide que la funcionalidad esté disponible provocará una prueba bloqueada, al igual que un entorno de prueba configurado incorrectamente.

## Formato de caso de prueba

La documentación del caso de prueba generalmente incluye toda la información pertinente para ejecutar y recopilar datos de la prueba. Si bien el formato de caso de prueba específico puede diferir entre organizaciones, la mayoría incluye los siguientes detalles:


- Nombre del módulo. Este es el módulo o característica bajo prueba.
- Identificación y/o nombre de la prueba. Este es un identificador único que debe seguir una convención de nomenclatura estándar.
- Nombre del tester. La persona que realiza la prueba.
- Datos de prueba. Esto describe los conjuntos de datos que se utilizarán para la prueba.
- Supuestos o condiciones previas. Describa los diversos pasos que se deben realizar antes de la prueba, o lo que podemos suponer situacionalmente sobre la prueba, como "después de un inicio de sesión exitoso".
- Prioridad de prueba. Defina si la prueba es de prioridad baja, media o alta.
- Escenarios de prueba. Como se describió anteriormente, esta es la acción de alto nivel de la que se deriva el caso de prueba.
- Entorno de prueba. Identificar el nombre y/o las características del entorno de prueba.
- Pasos de prueba. Detalle los pasos que debe seguir el tester en el orden deseado.
- Resultados previstos. Esta es la salida que espera recibir del sistema.
- Resultados actuales. Esta es la salida que realmente recibe del sistema.
- Determinación de pasa/falla. Si los resultados reales coinciden con los resultados esperados, la prueba pasa. Si no, la prueba falla.

La organización debe revisar, mantener y aprobar regularmente los casos de prueba para garantizar que cubran adecuadamente la funcionalidad nueva y antigua. Los

casos de prueba completamente detallados reducen la necesidad de pruebas exploratorias que requieren mucho tiempo para llenar los vacíos de cobertura.

### Modelo

ID	El ID se le da al caso de prueba.
Descripción	La descripción del caso de prueba.
Requisito relacionado	El ID se proporciona al requisito al que se asigna este caso de prueba.
Requisitos previos	Cualquier condición previa o requisito que debe cumplirse antes de ejecutar la prueba.
Pasos de la prueba	Se dieron instrucciones paso a paso para ejecutar la prueba.
Datos de prueba	Datos que se utilizan mientras se realiza la prueba.
Resultado esperado	El resultado que se espera de la prueba, registrado antes de ejecutar la prueba.
Resultado real	El resultado real obtenido después de ejecutar la prueba
Estado	El estado obtenido después de ejecutar la prueba. Puede ser Pasa, Falla, No Ejecutado, Bloqueado.
Comentarios	Cualquier comentario que se haga para la prueba.
Información del entorno	Incluye información de red/hardware/software en la que se ejecuta la prueba.

 **TIP:** Redacta casos de prueba de manera clara y concisa para asegurar precisión, independientemente del lector o ejecutor. Aunque algunos detalles son cruciales, busca mantener la economía y simplicidad en la ejecución para facilitar el mantenimiento ante posibles cambios en la aplicación. La reiterabilidad y reutilización son clave, ya que pocas pruebas se ejecutan sólo una vez, y las pruebas reutilizables optimizan el tiempo al desarrollar nuevas funciones. Asegúrate de que cada caso sea rastreable, facilitando la documentación y la presentación de resultados al equipo.

## Buenas prácticas para elaborar casos de prueba

La mejor práctica es desarrollar por lo menos dos casos de prueba para cada requerimiento a probar. A medida que se va conociendo el negocio y teniendo más experiencia en el testing, se va estimando mejor la cantidad de casos de prueba necesarios. Por ejemplo, para testear correcta y completamente un Login, requiere de 7 casos de prueba.

Los casos de prueba deben ser **simples**: se deben crear casos de prueba que sean lo más simple posible ya que otra persona que no sea el autor puede ejecutarlos. Usa un lenguaje asertivo como *“ir a la página de inicio”*, *“ingresar datos”*, *“hacer clic en esto”*, etc. Esto facilita la comprensión de los pasos de prueba y hace que la ejecución sea más rápida.

El **título** debe ser **fuerte**: la manera correcta de comenzar con el título de un caso de prueba es con un verbo en infinitivo. Los verbos en infinitivo denotan un mandato.

Tomar al **usuario final** en cuenta: el objetivo final es crear casos de prueba que cumplan con los requisitos del cliente y que sean fáciles de usar. Un tester debe crear casos de prueba tomando en cuenta la perspectiva del usuario final.

**Evitar la repetición** de casos de prueba: si se necesita un caso de prueba para ejecutar algún otro, llama al caso de prueba por su id. Inclúyelo en la columna de pre-condiciones o donde corresponda dependiendo la herramienta.

**No asumir:** No asumir la funcionalidad y las características de la aplicación mientras preparas el caso de prueba. Apégate a los documentos de especificación y si tienes alguna duda, pregunta.

Asegurar la **mayor cobertura** posible: Asegúrate de escribir casos de prueba para todos los requisitos especificados. Algo que se puede utilizar es una matriz de trazabilidad para garantizar que se prueben todos los casos de pruebas asociados. La mayoría de las herramientas permiten vincular los casos de prueba entre sí.

**Autonomía:** el caso de prueba debe generar los mismos resultados cada vez, sin importar quién lo pruebe.

Implementar **técnicas** de prueba: no es posible verificar todas las condiciones posibles de una aplicación, pero las técnicas de prueba ayudan a seleccionar los casos de prueba con la máxima posibilidad de encontrar un defecto.

Mantener un **repositorio estándar** de casos de prueba reutilizables para su aplicación garantizará que la mayoría de los errores comunes se detecten más rápidamente. La reutilización de los casos de prueba ayuda a ahorrar dinero en recursos para escribir pruebas repetitivas. Siempre se cubrirán casos de prueba importantes, por lo que será casi imposible olvidarlos.

Una **lista de verificación** ayuda a completar la redacción de casos de prueba rápidamente para nuevas versiones de la aplicación. Los casos de prueba se pueden consultar en la lista de verificación de pruebas para asegurarse de que los problemas más comunes se solucionen en la fase de desarrollo.

Definir la **prioridad** de una prueba. Generalmente es mejor utilizar cualquiera de los 3 niveles, alto, medio y bajo, o 1, 50 y 100. Por lo tanto, cuando tengas un cronograma, primero debes completar todas las pruebas de alta prioridad y luego pasar a las pruebas de prioridad media y baja. Para identificar estos niveles se debe conocer bien el negocio y la importancia de los niveles de error para el usuario.

## Factores de calidad de los casos de prueba

Un caso de prueba debe cumplir con los siguientes factores de calidad:

1. **Correcto.** Ser apropiado para los probadores y el entorno. Si teóricamente es razonable, pero exige algo que ninguno de los probadores tiene, se caerá por su propio peso.
2. **Exacto.** Demostrar que su descripción se puede probar.
3. **Económico.** Tener sólo los pasos o los campos necesarios para su propósito.
4. **Confiable y repetible.** Ser un experimento controlado con el que se obtiene el mismo resultado cada vez que se ejecuta, sin importar qué se pruebe.
5. **Rastreable.** Saber qué requisitos del caso de uso se prueban.
6. **Medible.** Este es un ejercicio muy útil para quienes escriben pruebas, para verificar constantemente dónde están, si pierden alguno de los elementos, o si no se cumple un estándar.