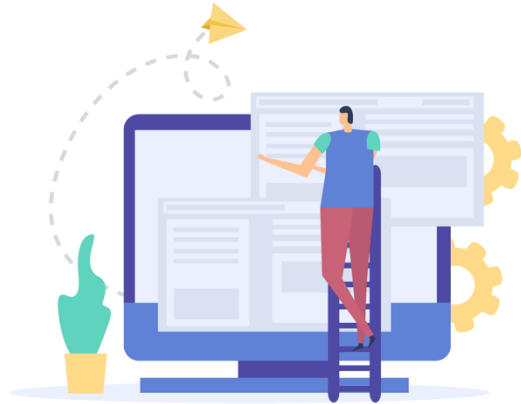


# Certificación de Pruebas de Software

La certificación del software se refiere al proceso de evaluar y verificar que el software cumple con ciertos estándares y requisitos específicos. La certificación del software se realiza normalmente por organizaciones independientes de pruebas de software, y puede ser requerida por los clientes o usuarios finales como una garantía de que el software es seguro y confiable.



Existen varias normas y estándares que se utilizan para la certificación del software, como ISO/IEC 12207, ISO/IEC 15504 (SPICE), ISO/IEC 9126 (calidad del software), ISO/IEC 25010 (calidad del producto de software), entre otros.

***El proceso de certificación del software puede incluir varias fases, tales como:***

1. **Evaluación de la documentación:** se revisa la documentación del software para asegurarse de que cumple con los estándares requeridos.
2. **Verificación del código:** se examina el código del software para asegurarse de que cumple con las buenas prácticas de programación y los estándares de calidad.
3. **Pruebas funcionales:** se realizan pruebas para comprobar que el software cumple con los requisitos funcionales y las especificaciones técnicas.
4. **Pruebas de seguridad:** se realizan pruebas para comprobar que el software es seguro y no presenta vulnerabilidades.
5. **Pruebas de rendimiento:** se realizan pruebas para comprobar que el software funciona correctamente bajo diferentes cargas de trabajo y situaciones.

Una vez que se ha completado el proceso de certificación, se emite un informe de certificación que documenta los resultados de las pruebas y verifica que el software cumple con los estándares y requisitos

requeridos. Este informe puede ser utilizado por los clientes o usuarios finales para tomar una decisión informada sobre la calidad y seguridad del software.

## EJEMPLO:

*Supongamos que una empresa desarrolla un software para una entidad financiera que permita la gestión de cuentas bancarias. Para asegurar que el software es seguro y confiable, se decide certificarlo de acuerdo con el estándar ISO/IEC 25010 para la calidad del producto de software.*

El proceso de certificación podría incluir las siguientes fases:

1. **Evaluación de la documentación:** se revisa la documentación del software, como el plan de pruebas, los requisitos funcionales y técnicos, y la documentación del código fuente, para asegurarse de que cumple con los estándares requeridos.
2. **Verificación del código:** se examina el código del software para asegurarse de que cumple con las buenas prácticas de programación y los estándares de calidad.
3. **Pruebas funcionales:** se realizan pruebas para comprobar que el software cumple con los requisitos funcionales y las especificaciones técnicas. Por ejemplo, se puede probar la funcionalidad de creación de cuentas, depósitos y retiros, transferencias de fondos, entre otras.
4. **Pruebas de seguridad:** se realizan pruebas para comprobar que el software es seguro y no presenta vulnerabilidades. Por ejemplo, se puede probar la seguridad de la autenticación y autorización, la privacidad de los datos y la prevención de ataques de hacking.
5. **Pruebas de rendimiento:** se realizan pruebas para comprobar que el software funciona correctamente bajo diferentes cargas de trabajo y situaciones. Por ejemplo, se puede probar el rendimiento del software bajo diferentes volúmenes de transacciones, usuarios concurrentes y picos de uso.

Un formato aplicable a este tipo de pruebas puede considerarse de acuerdo a la siguiente tabla:

No.	Requisito	Descripción	Estado
1.	Funcionalidad básica	Verificar que el software cumpla con todas las funcionalidades básicas especificadas en los requisitos del usuario	
2.	Compatibilidad del sistema	Verificar que el software sea compatible con todas las plataformas y sistemas operativos especificados en los requisitos del usuario	
3.	Seguridad	Verificar que el software cumpla con todos los requisitos de seguridad especificados en los requisitos del usuario, incluyendo la protección de datos y la prevención de ataques cibernéticos	
4.	Usabilidad	Verificar que el software sea fácil de usar y cumpla con todos los requisitos de usabilidad especificados en los requisitos del usuario	
5.	Rendimiento	Verificar que el software cumpla con todos los requisitos de rendimiento especificados en los requisitos del usuario, incluyendo los tiempos de respuesta y la capacidad de procesamiento de datos	
6.	Mantenibilidad	Verificar que el software sea fácil de mantener y actualizar, y cumpla con todos los requisitos de mantenibilidad especificados en los requisitos del usuario	
7.	Documentación	Verificar que el software tenga una documentación completa y clara, incluyendo manuales de usuario y manuales técnicos	

*Una vez que se ha completado el proceso de certificación, se emite un informe de certificación que documenta los resultados de las pruebas y verifica que el software cumple con los estándares y requisitos requeridos. Este informe puede ser utilizado por la entidad financiera para tomar una decisión informada sobre la calidad y seguridad del software.*

Un informe de certificación para un software podría ser de la siguiente manera:

#### **Informe de Certificación del Software XYZ de la Entidad Financiera ABC**

##### **Introducción**

El objetivo de este informe es documentar los resultados de las pruebas de certificación del software XYZ de la Entidad Financiera ABC. El software fue sometido a pruebas exhaustivas para verificar que cumpla con los estándares y requisitos de la entidad financiera.

##### **Resultados de las Pruebas**

Las pruebas fueron realizadas por el equipo de certificación de la entidad financiera ABC durante un período de dos semanas. El software fue sometido a pruebas en los siguientes aspectos:

1. **Funcionalidad básica:** se verificó que el software cumpla con todas las funcionalidades básicas especificadas en los requisitos del usuario. El software cumplió con todas las funcionalidades básicas especificadas.
2. **Compatibilidad del sistema:** se verificó que el software sea compatible con todas las plataformas y sistemas operativos especificados en los requisitos del usuario. El software cumplió con todas las plataformas y sistemas operativos especificados.
3. **Seguridad:** se verificó que el software cumpla con todos los requisitos de seguridad especificados en los requisitos del usuario, incluyendo la protección de datos y la prevención de ataques cibernéticos. El software cumplió con todos los requisitos de seguridad especificados.
4. **Usabilidad:** se verificó que el software sea fácil de usar y cumpla con todos los requisitos de usabilidad especificados en los requisitos del usuario. El software cumplió con todos los requisitos de usabilidad especificados.
5. **Rendimiento:** se verificó que el software cumpla con todos los requisitos de rendimiento especificados en los requisitos del usuario, incluyendo los tiempos de respuesta y la capacidad de procesamiento de datos. El software cumplió con todos los requisitos de rendimiento especificados.
6. **Mantenibilidad:** se verificó que el software sea fácil de mantener y actualizar, y cumpla con todos los requisitos de mantenibilidad especificados en los requisitos del usuario. El software cumplió con todos los requisitos de mantenibilidad especificados.
7. **Documentación:** se verificó que el software tenga una documentación completa y clara, incluyendo manuales de usuario y manuales técnicos. El software cumplió con todos los requisitos de documentación especificados.

### Conclusión

Basado en los resultados de las pruebas, el software XYZ de la Entidad Financiera ABC cumple con todos los estándares y requisitos especificados en los requisitos del usuario. Por lo tanto, se recomienda que la entidad financiera lo use para sus operaciones diarias.

*Firma del Equipo de Certificación:*

Fecha: \_\_\_\_\_

## Diferencias entre Metodologías Tradicionales y Ágiles en el desarrollo de software

### Waterfall



### Agile/Iterative



### *Metodología Tradicional*

En una metodología tradicional de desarrollo de software, como por ejemplo el modelo en cascada o el modelo en V, el proceso de certificación del software suele realizarse al final del ciclo de vida del software, después de que se ha completado el desarrollo y antes de la entrega del software al cliente.

El proceso de certificación del software en una metodología tradicional de desarrollo de software puede incluir las siguientes fases:

1. **Planificación de la certificación:** se define un plan de certificación que establece los estándares y requisitos que el software debe cumplir, así como los criterios y métodos de evaluación que se utilizarán.
2. **Diseño de las pruebas:** se diseña un conjunto de pruebas que se utilizarán para evaluar el software, incluyendo pruebas funcionales, pruebas de seguridad, pruebas de rendimiento y otras pruebas relevantes.
3. **Implementación de las pruebas:** se implementan las pruebas diseñadas y se llevan a cabo las pruebas del software.
4. **Análisis de resultados y corrección de errores:** se analizan los resultados de las pruebas y se corrigen los errores y problemas encontrados en el software.
5. **Certificación:** una vez que se ha completado el proceso de pruebas y corrección de errores, se emite un informe de certificación que documenta los resultados de las pruebas y verifica que el software cumple con los estándares y requisitos requeridos.

En resumen:

*Es importante tener en cuenta que el proceso de certificación del software en una metodología tradicional de desarrollo de software puede ser costoso y prolongado, ya que las pruebas se llevan a cabo al final del ciclo de vida del software y pueden requerir cambios significativos en el software. Para evitar esto, muchas empresas están adoptando metodologías de desarrollo ágil que permiten realizar pruebas de forma continua durante todo el ciclo de vida del software, lo que puede mejorar la calidad del software y reducir los costos y tiempos de certificación.*

## **Metodología Ágil**

En una metodología ágil, el proceso de certificación del software se integra en el ciclo de vida del software de forma continua durante todo el proceso de desarrollo. Esto permite detectar y corregir los errores y problemas de calidad del software de manera temprana, lo que puede reducir significativamente los costos y tiempos de certificación.

El proceso de certificación del software en una metodología ágil podría incluir las siguientes fases:

1. **Definición de requisitos de calidad:** se definen los requisitos de calidad que el software debe cumplir, incluyendo los estándares y requerimientos de la industria, los objetivos de calidad específicos y las expectativas del cliente.
2. **Diseño de pruebas:** se diseña un conjunto de pruebas que se utilizarán para evaluar el software, incluyendo pruebas funcionales, pruebas de seguridad, pruebas de rendimiento y otras pruebas relevantes. Estas pruebas se diseñan de manera iterativa a lo largo del ciclo de vida del software, a medida que se van completando las funcionalidades del software.
3. **Implementación de pruebas automatizadas:** se implementan pruebas automatizadas que permiten realizar pruebas continuas del software a medida que se desarrolla. Estas pruebas se ejecutan automáticamente cada vez que se realizan cambios en el software, lo que permite detectar y corregir errores de manera temprana.
4. **Análisis de resultados y corrección de errores:** se analizan los resultados de las pruebas y se corrigen los errores y problemas encontrados en el software. Este proceso se lleva a cabo de forma continua a medida que se desarrolla el software.
5. **Certificación:** una vez que se ha completado el proceso de pruebas y corrección de errores, se emite un informe de certificación que documenta los resultados de las pruebas y verifica que el software cumple con los estándares y requisitos requeridos.

En Resumen:

*En una metodología ágil, el proceso de certificación del software se integra de forma continua durante todo el ciclo de vida del software, lo que permite detectar y corregir errores de manera temprana y reducir los costos y tiempos de certificación. Esto se logra mediante el diseño y la implementación de pruebas automatizadas, así como la integración continua y la entrega continua del software.*

# Formatos

Existen diferentes formatos de pruebas de software que se utilizan para documentar y registrar el proceso de pruebas y los resultados obtenidos. A continuación, se presentan algunos de los formatos más comunes:

1. **Casos de prueba:** los casos de prueba son documentos que describen detalladamente los pasos que deben seguirse para ejecutar una prueba específica. Los casos de prueba incluyen información sobre las entradas y salidas esperadas, así como los criterios de aceptación y rechazo de la prueba.

## **Caso de prueba #1:**

Nombre: Verificar la funcionalidad de ingreso de nombre y apellido

Objetivo: Verificar que el usuario pueda ingresar su nombre y apellido correctamente en el formulario de registro.

Pasos:

- a. Acceder al formulario de registro
- b. Ingresar un nombre y apellido válido en los campos correspondientes
- c. Hacer clic en el botón "Guardar"
- d. Verificar que el nombre y apellido ingresados se muestren correctamente en la pantalla de confirmación

Criterios de aceptación:

- El nombre y apellido ingresados se muestran correctamente en la pantalla de confirmación
- El sistema no permite ingresar caracteres inválidos en los campos de nombre y apellido
- El sistema permite ingresar nombres y apellidos con caracteres especiales y espacios en blanco

## **Caso de prueba #2:**

Nombre: Verificar la funcionalidad de ingreso de correo electrónico

Objetivo: Verificar que el usuario pueda ingresar su correo electrónico correctamente en el formulario de registro.

Pasos:

- a. Acceder al formulario de registro
- b. Ingresar un correo electrónico válido en el campo correspondiente
- c. Hacer clic en el botón "Guardar"
- d. Verificar que el correo electrónico ingresado se muestre correctamente en la pantalla de confirmación

Criterios de aceptación:

- El correo electrónico ingresado se muestra correctamente en la pantalla de confirmación
- El sistema valida que el correo electrónico tenga un formato válido
- El sistema no permite ingresar correos electrónicos duplicados



**Caso de prueba #3:**

Nombre: Verificar la funcionalidad de ingreso de contraseña

Objetivo: Verificar que el usuario pueda ingresar su contraseña correctamente en el formulario de registro.

Pasos:

- a. Acceder al formulario de registro
- b. Ingresar una contraseña válida en el campo correspondiente
- c. Hacer clic en el botón "Guardar"
- d. Verificar que la contraseña ingresada se muestre correctamente en la pantalla de confirmación

Criterios de aceptación:

- La contraseña ingresada se muestra correctamente en la pantalla de confirmación
- El sistema valida que la contraseña tenga una longitud mínima y cumpla con los requisitos de seguridad
- El sistema encripta la contraseña antes de almacenarla en la base de datos

2. **Plan de pruebas:** el plan de pruebas es un documento que describe la estrategia de pruebas a seguir para evaluar el software. El plan de pruebas incluye información sobre el alcance de las pruebas, las herramientas de pruebas utilizadas, el equipo de pruebas, los recursos necesarios y los plazos.

**Plan de pruebas del sistema de compras en línea****a. Objetivo:**

El objetivo de este plan de pruebas es evaluar la funcionalidad del sistema de compras en línea para asegurar que cumple con los requisitos especificados en la documentación del proyecto.

**b. Alcance:**

- Este plan de pruebas cubrirá las siguientes funcionalidades del sistema de compras en línea:
- Registro de usuarios
- Inicio de sesión
- Búsqueda de productos
- Añadir productos al carrito de compras
- Proceso de pago
- Historial de compras

**c. Estrategia de pruebas:**

Las pruebas serán realizadas en un ambiente de pruebas separado del ambiente de producción. Se utilizará una combinación de pruebas manuales y automatizadas para evaluar el sistema. Las pruebas manuales serán realizadas por el equipo de pruebas y las pruebas automatizadas serán ejecutadas por herramientas de prueba automatizadas.

**d. Plan de pruebas:**

- a. Pruebas de registro de usuarios:
  - Verificar que el formulario de registro esté funcionando correctamente
  - Verificar que los usuarios puedan registrarse exitosamente
  - Verificar que el sistema no permita el registro de usuarios con correo electrónico duplicado

**b. Pruebas de inicio de sesión:**

- Verificar que el formulario de inicio de sesión esté funcionando correctamente
- Verificar que los usuarios puedan iniciar sesión exitosamente
- Verificar que el sistema no permita iniciar sesión con credenciales inválidas

**c. Pruebas de búsqueda de productos:**

- Verificar que los resultados de búsqueda sean precisos
- Verificar que los filtros de búsqueda funcionen correctamente
- Verificar que el sistema no muestre productos fuera de stock

**d. Pruebas de añadir productos al carrito de compras:**

- Verificar que los productos sean añadidos correctamente al carrito
- Verificar que el sistema actualice el precio total del carrito correctamente

**e. Pruebas del proceso de pago:**

- Verificar que los usuarios puedan ingresar su información de pago correctamente
- Verificar que el sistema procese los pagos correctamente
- Verificar que los usuarios reciban confirmación de la compra después del proceso de pago

**f. Pruebas del historial de compras:**

- Verificar que los usuarios puedan ver su historial de compras
- Verificar que el historial de compras muestre información precisa y detallada

**e. Criterios de aceptación:**

El sistema será considerado aceptado si se cumplen los siguientes criterios:

- Todos los casos de prueba han sido ejecutados y documentados correctamente
- Todos los problemas detectados durante las pruebas han sido resueltos
- El sistema cumple con todos los requisitos especificados en la documentación del proyecto

3. **Informe de pruebas:** el informe de pruebas es un documento que registra los resultados de las pruebas realizadas. El informe de pruebas incluye información sobre las pruebas realizadas, los errores encontrados, los resultados de las pruebas y las recomendaciones para la corrección de errores.

## EJEMPLO:

### Informe de pruebas del sistema de gestión de inventario

**a. Resumen ejecutivo:**

El sistema de gestión de inventario ha sido probado exhaustivamente por el equipo de pruebas. Se han encontrado algunos problemas durante las pruebas, pero la mayoría de ellos se han resuelto y el sistema cumple con los requisitos del proyecto.

**b. Resultados de las pruebas:**

Durante las pruebas, se encontraron los siguientes problemas:

Funcionalidad	Casos de Prueba	Prioridad	Estado
Inicio de sesión	El sistema permite iniciar sesión sin autenticación	Alta	Corregido
Registro de productos	El sistema no valida los campos del formulario de registro de productos	Media	Pendiente
Búsqueda de productos	El filtro de búsqueda por precio no funciona correctamente	Baja	Pendiente

**c. Cobertura de las pruebas:**

Se han cubierto todas las funcionalidades del sistema, incluyendo el registro de usuarios, inicio de sesión, registro de productos, búsqueda de productos y generación de reportes.

**d. Herramientas utilizadas:**

Se han utilizado las siguientes herramientas durante las pruebas:

- Selenium para pruebas automatizadas de la interfaz de usuario
- JMeter para pruebas de carga y rendimiento
- Postman para pruebas de API

**e. Conclusiones:**

En general, el sistema de gestión de inventario ha pasado la mayoría de las pruebas con éxito. Los problemas encontrados se han resuelto y se han documentado para futuras mejoras. Se recomienda realizar pruebas adicionales antes del lanzamiento del sistema para asegurar que no se presenten nuevos problemas.

**f. Recomendaciones:**

Se recomienda realizar las siguientes acciones:

- Realizar pruebas adicionales de validación del formulario de registro de productos
- Solucionar el filtro de búsqueda por precio para mejorar la precisión de los resultados

- Realizar pruebas adicionales de carga y rendimiento para verificar la capacidad del sistema bajo cargas pesadas

4. **Registro de incidentes:** el registro de incidentes es un documento que registra los errores y problemas encontrados durante las pruebas. El registro de incidentes incluye información sobre la descripción del problema, la gravedad del problema, la fecha de detección y la persona responsable de la resolución del problema.

### Registro de incidentes del sistema de compras en línea

No. Incidente	Fecha	Funcionalidad	Descripción del incidente	Prioridad	Estado
001	01/02/2023	Proceso de pago	El sistema no permite realizar el pago con tarjeta de crédito	Alta	En proceso de resolución
002	02/02/2023	Registro de usuarios	El sistema no permite el registro de usuarios con correo electrónico duplicado	Media	Resuelto
003	02/02/2023	Búsqueda de productos	El filtro de búsqueda por precio no funciona correctamente	Baja	Pendiente
004	03/02/2023	Proceso de envío	Los usuarios no reciben una confirmación de envío después de realizar la compra	Media	En proceso de resolución
005	04/02/2023	Proceso de pago	El sistema no muestra el monto total de la compra antes de realizar el pago	Baja	Resuelto

En este registro de incidentes, se ha documentado la información relevante sobre cada incidente encontrado durante las pruebas del sistema de compras en línea. Se ha registrado un número de incidente único para cada problema, junto con la fecha de reporte, la funcionalidad afectada, una descripción detallada del incidente, su prioridad y estado actual.

Con este registro de incidentes, se puede hacer un seguimiento de los problemas encontrados en el sistema, su estado de resolución y la prioridad de cada problema para ayudar al equipo de desarrollo a priorizar sus esfuerzos y mejorar la calidad del sistema.

5. **Lista de verificación de pruebas:** la lista de verificación de pruebas es un documento que contiene una lista de criterios de calidad que deben verificarse durante las pruebas. La lista de verificación de pruebas se utiliza para asegurar que se evalúan todos los aspectos relevantes del software.

Lista de verificación de pruebas para sitio web de comercio electrónico

No.	Funcionalidad	Prueba	Estado
1.	Funcionalidad de inicio de sesión	Verificar que los usuarios puedan iniciar sesión con éxito utilizando sus credenciales	
		Verificar que los usuarios no puedan iniciar sesión con credenciales incorrectas	
		Verificar que el sistema maneje de manera adecuada el cierre de sesión de usuarios	
2.	Funcionalidad de búsqueda	Verificar que la búsqueda de productos por nombre funcione correctamente	
		Verificar que la búsqueda de productos por categoría funcione correctamente	
		Verificar que los resultados de búsqueda sean precisos y relevantes	
3.	Funcionalidad de registro de usuario	Verificar que los usuarios puedan registrarse con éxito proporcionando la información requerida	
		Verificar que el sistema maneje de manera adecuada el registro de usuarios con información duplicada	
		Verificar que el sistema envíe un correo electrónico de confirmación después de que los usuarios se registren	
4.	Funcionalidad de carrito de compras	Verificar que los usuarios puedan agregar productos al carrito de compras	
		Verificar que los usuarios puedan modificar la cantidad de productos en el carrito de compras	
		Verificar que los usuarios puedan eliminar productos del carrito de compras	
5.	Funcionalidad de proceso de pago	Verificar que los usuarios puedan seleccionar un método de pago y proporcionar la información necesaria	
		Verificar que el sistema calcule correctamente el monto total de la compra, incluyendo impuestos y gastos de envío	
		Verificar que el sistema muestre una confirmación de compra después de que los usuarios completen el proceso de pago	

En esta tabla, el equipo de pruebas puede registrar el estado de cada prueba como "completa" o "pendiente", para que sea más fácil hacer un seguimiento del progreso de las pruebas.

Estos son solo algunos de los formatos de pruebas de software que se utilizan comúnmente en el campo de la ingeniería de software. La elección del formato de pruebas adecuado dependerá del objetivo de las pruebas, el contexto y la metodología de desarrollo utilizada.