

Nivel de Pruebas

Introducción

La implementación de pruebas desempeña un papel esencial en el desarrollo de software, siendo fundamental para asegurar la calidad y confiabilidad de un sistema. Estas pruebas se llevan a cabo en diversos niveles y se diseñan con objetivos específicos en mente. A continuación, exploraremos en detalle los cuatro niveles típicos de pruebas: pruebas unitarias o de componentes, pruebas de integración, pruebas de sistema y pruebas de aceptación.

Veamos cada uno en profundidad:

1. Pruebas unitarias o de componentes (unit testing / component testing)

Este tipo de pruebas suelen ser creadas por los desarrolladores en paralelo con el desarrollo del código para la funcionalidad en la que se esté trabajando en ese momento.

Las pruebas unitarias tienen como objetivo verificar que un componente o unidad específica dentro del código, funcione de manera correcta.

El componente bajo test puede ser tanto funcional como no funcional.

Una prueba unitaria podría crearse para validar que cierto método específico en el código devuelve un valor "X".

Los insumos para las pruebas unitarias son:

- el detalle del diseño
- las especificaciones del componente bajo prueba (en caso de que existan)
- el mismo código escrito.

En general este tipo de pruebas se automatizan y conforman luego, las pruebas de regresión.

2. Pruebas de integración (integration testing)

Este tipo de pruebas son creadas por los desarrolladores con foco en probar las interfaces y la interacción e integración entre los distintos componentes del sistema.

Las pruebas de integración se podrían clasificar en:

- **Pruebas de integración de componentes** (component integration testing): son aquellas pruebas de interacciones e interfaces entre componentes integrados.
- **Pruebas de integración de sistema** (system integration testing): son aquellas pruebas de interacciones e interfaces entre sistemas, paquetes, microservicios y con sistemas externos.

En este nivel los objetos de prueba podrían ser: subsistemas, bases de datos, infraestructura, interfaces, APIs (application program interfaces) y microservicios. Estas pruebas también se automatizan y se agregan a las pruebas de regresión.

3. Pruebas de sistema (system testing)

Este tipo de pruebas consideran la funcionalidad o sistema bajo análisis desde una perspectiva E2E (end to end). Se enfocan en el comportamiento del sistema o producto en su totalidad, tal como fue descrito en la documentación (base para el diseño de pruebas)

Estas pruebas no son llevadas a cabo por los desarrolladores ni se basan en cómo está escrito el código. Este tipo de pruebas se planifican y ejecutan por el equipo de testing.

Las pruebas de sistema incluyen pruebas funcionales y no funcionales

Objetivos de las pruebas de sistema:

- Verificar que el sistema creado cumpla con los requerimientos funcionales y no funcionales que fueron especificados para su construcción.
- Validar que el sistema esté completo y que funcione de la manera esperada.
- Encontrar defectos y reducir riesgo de fallos del sistema en producción.
- Poner de manifiesto el nivel de calidad del sistema.

4. Pruebas de aceptación (acceptance testing)

El objetivo es probar que el sistema funciona acorde a lo solicitado por el usuario final tanto a nivel funcional como operacional.

Las pruebas de aceptación se pueden clasificar en:

- **Pruebas de aceptación del usuario** (UAT – user acceptance testing). En estas, se solicita al usuario final (o un representante del grupo de usuarios finales) que pruebe el sistema y de el visto bueno final confirmando que el sistema cumple con lo esperado.
- **Pruebas de aceptación operacional:** Aquí, se prueban los procesos y procedimientos para que el sistema pueda ser usado y operable. Esto podría incluir pruebas de: performance, de seguridad o pentesting, back up, actualizaciones, tareas de carga de datos y migraciones, procedimientos de mantenimiento, procedimientos de recuperación.
- **Pruebas de aceptación contractuales:** Cuando el criterio de aceptación está documentado en un contrato, las pruebas se llevan a cabo para confirmar que lo descrito haya sido diseñado e implementado correctamente
- **Pruebas de aceptación regulatorias:** Cuando el sistema creado -por su naturaleza- se encuentra bajo regulaciones o estándares gubernamentales, legales o de seguridad, las pruebas deben ser auditadas por los grupos regulatorios correspondientes.
- **Pruebas en Alpha** (Alpha testing): El sistema se prueba en su versión Alpha. Desde la experiencia del desarrollador
- **Pruebas en beta** (Beta testing): El sistema se prueba desde la experiencia del cliente antes de la versión estable.

📌 El testing en **Alpha** y en **Beta** se usa (sobre todo) para recibir feedback antes del “go live” () final

El objetivo final de las pruebas de aceptación no solo encontrar defectos, si no hacer un chequeo final y comprobar que el sistema se ajusta a las necesidades del negocio y a los requerimientos comunicados.



💡 **Información de la industria :**

Probablemente el trabajo de análisis, planificación y ejecución de pruebas (anterior a la etapa de las pruebas de aceptación) ofrecería un sistema libre de bugs y ajustado a lo solicitado. Sin embargo, si se encuentran defectos, deben reportarse, pero en general las pruebas de aceptación son responsabilidad del cliente.