

Tipos de Prueba de Software

Diferentes aplicaciones dependiendo de cuál sea el objetivo de la misma.

Las pruebas de software contribuyen a **aumentar la calidad y la confianza**, el producto que se les presenta ofrece la garantía de haber sido testeado en todas las etapas de su desarrollo. Resolver un incidente durante el desarrollo siempre es menos costoso que tener que asumirlo una vez que el software está ya en producción.

Pruebas unitarias o unit testing

Comprueban que **cada una de las piezas** o unidades más pequeñas **del software** en el que se está trabajando **funcione correctamente**.

Estas pruebas se aplican de manera individual y son las primeras que deben realizarse durante todo el proceso de desarrollo.

Es necesario aislar esas unidades, que pueden ser fragmentos de código, para verificar su comportamiento.

Son muy rápidas y facilitan la comprobación de la legibilidad del código.

Ventajas

Al trabajar con unidades tan pequeñas es posible testear el proyecto por partes, sin necesidad de que esté terminado.

Entre sus ventajas está el ahorro de tiempo y de dinero, ya que permiten detectar errores al principio y evitar seguir avanzando arrastrando ese fallo que va a condicionar el resto del proyecto.

Pruebas de integración

Cualquier aplicación de software está formada por varios elementos que deben encajar a la perfección.

El objetivo de estas pruebas es comprobar que los diferentes componentes operan bien juntos.

En ocasiones sucede que un módulo que funcionaba perfectamente de manera aislada, una vez que se **integra** con el resto provoca algún tipo de incidente, por lo que es necesario probar esa integración para obtener los mejores resultados.

Es un proceso más complejo y debe realizarse después de las pruebas unitarias.

Pruebas funcionales

Lo que se comprueba son las funciones del software creado para establecer la **usabilidad**, el comportamiento del sistema, si todo funciona tal y como está establecido en la documentación y los requerimientos del software.

Ejemplo: pruebas de caja negra o black-box testing, pruebas unitarias y de regresión

Pruebas de aceptación

Antes de comenzar a trabajar, el equipo responsable del proyecto debe definir con estas pruebas **es si el software que se ha creado es lo que el cliente y los usuarios esperaban**. Por otra parte, si durante la fase de desarrollo se toma cualquier decisión que añada o disminuya criterios, también debe dejarse constancia.

Durante las pruebas unitarias y de integración se realizarán las pruebas de aceptación para corroborar que todo el sistema funciona según lo esperado.

Pruebas de rendimiento

Se trata de chequear la **respuesta del software ante cargas de trabajo diferentes y en condiciones reales**. Sirven para definir cuestiones como la estabilidad o la velocidad de una aplicación de software y están directamente relacionadas con la experiencia del usuario y la tasa de conversión.

Las pruebas de carga, de resistencia, estrés, volumen, picos y escalabilidad son consideradas pruebas de rendimiento.

Pruebas de estrés

Antes de dar por finalizado el proceso de desarrollo de una software es necesario **comprobar cuánta tensión puede soportar antes de que se produzca algún error**. Para ello, se envía mucha más información de lo habitual para comprobar en qué punto se satura el sistema.

Pruebas de regresión

En ocasiones, el desarrollador decide modificar alguna funcionalidad en el sistema. Este tipo de pruebas tienen como objetivo **comprobar que los cambios en un componente del software no provocan reacciones no deseadas** o fallos en otros elementos no modificados. No tendría sentido introducir una nueva función si esto va a tener consecuencias negativas sobre el resto.

Pruebas de humo

Son pruebas funcionales que ayudan a determinar si el conjunto del software funciona bien y si está preparado para ser sometido a pruebas más exhaustivas. Están pensadas para **garantizar la funcionalidad principal**, así que si no se superan no se avanza ni se programa ninguna prueba más compleja. Hay que volver al punto de inicio.