

Pruebas de Software

Las pruebas de software son esenciales para garantizar la calidad, detectar errores antes de la implementación y mejorar la experiencia del usuario. Ayudan a reducir costos y tiempos de mantenimiento al identificar fallos en etapas tempranas del desarrollo.

Planificación de las Pruebas

¿Por qué son necesarias?

Las pruebas aseguran que el software funcione correctamente, minimizando riesgos y mejorando su estabilidad.

Organización de las pruebas

El proceso de planificación incluye:

- ✓ Identificación de requisitos del software.
 - ✓ Definición del alcance y objetivos de las pruebas.
 - ✓ Selección de herramientas y metodologías.
 - ✓ Creación de cronogramas y asignación de recursos.
-

Tipos de Pruebas

Caja Negra

Evalúa la funcionalidad sin analizar el código interno. Se basa en los requisitos y casos de uso.

Caja Blanca

Examina la estructura interna del código, validando flujos lógicos, ciclos y condiciones.

Pruebas de Carga

Analizan el rendimiento del sistema bajo condiciones específicas.

Pruebas de Estrés

Evalúan el comportamiento del software en situaciones extremas, como picos de tráfico o uso intensivo.

Pruebas de Estabilidad

Comprueban el rendimiento del software tras largos períodos de uso continuo.

Pruebas de Regresión

Verifican que nuevas modificaciones en el código no afecten funcionalidades previas.

Pruebas Estructurales

Evalúan la integración y arquitectura del software.

Pruebas Funcionales

Validan que cada función del software opere conforme a los requisitos definidos.

Procedimientos y Casos de Prueba

¿Cómo se diseñan y aplican?






El diseño de pruebas sigue estos pasos:

- 1 **Identificación de requisitos** y criterios de aceptación.
 - 2 **Creación de escenarios de prueba** basados en casos reales.
 - 3 **Ejecución y registro de resultados.**
 - 4 **Análisis de fallos y corrección** de errores.
 - 5 **Repetición de pruebas** para validar correcciones.
-

Herramientas de Depuración

¿Qué son y cómo ayudan?

Las herramientas de depuración permiten detectar y corregir errores en el código. Algunas de las más utilizadas son:

Herramienta	Descripción
 GDB	Depuración de programas en C/C++.
 Visual Studio Debugger	Depuración en entornos Microsoft.
 Postman	Pruebas de APIs.
 Selenium	Pruebas automatizadas en aplicaciones web.
 JMeter	Pruebas de carga y rendimiento.

Validaciones






Importancia del Cliente en la Evaluación


El cliente juega un papel clave en la validación del software. A través de las **pruebas de aceptación del usuario (UAT)**, se verifica que el producto cumpla con sus expectativas antes de su lanzamiento.

Normas de Calidad

Principales estándares en la industria

Las normas de calidad establecen directrices para el desarrollo de software eficiente y confiable:

Norma	Descripcion	
-----	 ISO/IEC 25010 Define características de calidad del software.	 CMMI
 ISO/IEC 29119 Normas para pruebas de software.	 ISTQB Buenas prácticas en pruebas de software.	
 IEEE 829 Documentación de pruebas de software.		

 **Conclusión:** Implementar pruebas adecuadas mejora la calidad, estabilidad y eficiencia del software, reduciendo errores y asegurando su correcto funcionamiento antes del despliegue.