

Planificación de las Pruebas

¿Por qué son necesarias?

Las pruebas verifican que el software en desarrollo sea correcto y cumpla con las características exigidas por el cliente.

¿Cómo se organizan?

Se implementa una estrategia de pruebas, como el **Modelo en Espiral**, siguiendo este orden:

- 1 Pruebas unitarias
 - 2 Pruebas de integración
 - 3 Pruebas de validación
 - 4 Pruebas de sistema
-

Tipos de Pruebas

♦ Caja Negra

Verifica que la aplicación devuelva los resultados correctos según las entradas recibidas.

♦ Caja Blanca

Evalúa la lógica interna de la aplicación, no solo el resultado final sino también el proceso.

♦ Pruebas de carga

Analizan el comportamiento del software ante un alto número de usuarios concurrentes. Tipos principales:

- **Prueba de estrés:** Determina el máximo número de usuarios soportados antes de que el sistema falle.
- **Prueba de estabilidad:** Evalúa si la aplicación soporta la carga esperada de forma continua.
- **Prueba de picos:** Simula cambios drásticos en la carga del sistema.

♦ Pruebas de regresión

Detectan nuevos errores, divergencias funcionales o carencias de funcionalidad tras modificaciones en el software.

- ♦ **Pruebas funcionales**

Verifican si el software cumple con sus requerimientos.

- ♦ **Pruebas estructurales**

Analizan la arquitectura interna del software.

Procedimientos y Casos de Prueba

¿Cómo se diseñan y aplican?

Para asegurar una detección efectiva de errores, se pueden seguir distintos enfoques:

- **Caja negra:** Evalúa que el programa produzca los resultados esperados.
 - **Caja blanca:** Verifica que el código interno siga las especificaciones necesarias.
 - **Enfoque aleatorio:** Se prueban diferentes combinaciones de entrada.
-

Herramientas de Depuración

¿Qué son y cómo ayudan a detectar errores?

- ♦ **Errores de compilación:** Fallos sintácticos (como falta de ; o paréntesis). El entorno de desarrollo suele identificarlos automáticamente.

- ♦ **Errores de lógica o bugs:** No impiden la compilación, pero generan valores incorrectos o cierres inesperados del programa.


Para solucionar estos errores, se utilizan **depuradores**, herramientas que supervisan la ejecución del código para localizar y corregir fallos lógicos.


Validaciones

Importancia del Cliente en la Evaluación del Software

El cliente juega un papel clave en la validación mediante pruebas de **caja negra** para verificar el cumplimiento de requisitos.

Posibles resultados de una prueba

 **Aceptable:** Cumple con las especificaciones y es funcional.

 **Deficiencias:** Se detectan desviaciones de las especificaciones, generando una lista de errores a corregir.

Normas de Calidad

Principales Estándares Utilizados en la Industria

◆ Estándares BSI:

- **BS 7925-1:** Vocabulario de pruebas de software.
- **BS 7925-2:** Pruebas de los componentes de software.

◆ Estándares IEEE:

- **IEEE 829:** Documentación de pruebas de software.
- **IEEE 1008:** Pruebas de unidad.

◆ Otros estándares:

- **ISO/IEC 12207, 15289, 29119** (gestión, organización y procesos de prueba).

Estos estándares establecen procedimientos, pero no siempre cubren la organización y gestión de pruebas, por lo que se han desarrollado normativas específicas como **ISO/IEC 29119**.