

PRUEBA DE SOFTWARE

Probar bien un sistema no es una actividad trivial para aprender. Algunos lo consideran un arte y aprender a hacerlo bien requiere práctica y experiencia. El 50% del tiempo y esfuerzo del desarrollo de Software corresponde a la prueba.

LA PRUEBA DE UN SISTEMA

Se define como el proceso de ejercitar o evaluar el sistema, por medios manuales o automáticos, para verificar que satisface los requerimientos o, para identificar diferencias entre los resultados esperados y los que produce el sistema (IEEE).

ASPECTOS IMPORTANTES SOBRE PRUEBA Y DEPURACIÓN

PRUEBA

Es el proceso mediante el cual se establece la existencia de errores.

DEPURACIÓN

Es el proceso mediante el cual se localizan los errores.

Quién debe hacer la prueba? Se recomienda conformar un equipo con:

- ♦ Un moderador (persona experimentada e independiente del proyecto).
- ♦ El diseñador del programa.
- ♦ El programador.
- ♦ Un responsable de la prueba.

Qué debe considerarse para la realización de la prueba?

- ♦ Disponibilidad de herramientas del lenguaje para facilitar la prueba y depuración del sistema.
 - Generadores de datos para el test;
 - Herramientas del computador que producen resumen del flujo y número de ejecuciones de las instrucciones de una aplicación;
 - Comparadores de archivos.
- ♦ Técnicas para efectuar la prueba.
- ♦ La depuración del sistema.

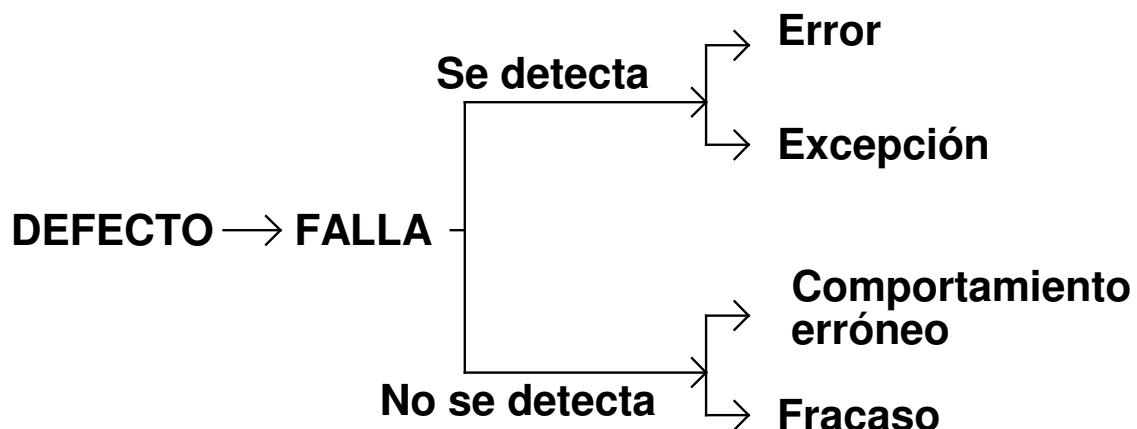
**La prueba puede demostrar existencia de errores,
NO LA AUSENCIA DE ELLOS**

PROCEDIMIENTOS PARA PRUEBA DE SOFTWARE

DEFECTO en un ambiente, algoritmo o dato, cuando esa entidad no reúne sus especificaciones.

FALLA puede resultar cuando se persiste en mantener una entidad con un defecto.

Cuando el sistema procesa una falla se da lugar a comportamientos anormales conocidos por: **error, excepción, comportamiento erróneo y fracaso.**



ERROR

Cuando el sistema detecta una falla, la reconoce y maneja de tal forma que el procesamiento normal puede continuar.

EXCEPCIÓN

Cuando el sistema encuentra una falla, la reconoce pero no la puede manejar de forma que el proceso normal pueda continuar.

COMPORTAMIENTO ERRONEO

Cuando el sistema encuentra una falla que no es reconocible y que no causa una violación observable de sus especificaciones.

FRACASO

Cuando el sistema encuentra una falla que no es reconocible y que causa que se violen las especificaciones.

VERIFICACIÓN

Intento de encontrar defectos ejecutando el programa en un ambiente de prueba.

VALIDACIÓN

Intento de encontrar defectos ejecutando el programa en un ambiente real.

NIVELES DE PRUEBA

- ♦ Prueba de módulo o de unidad.
- ♦ Prueba de integración.
- ♦ Prueba funcional.
- ♦ Prueba del sistema y prueba de aceptación.
- ♦ Prueba de instalación.
- ♦ Prueba de vida.

TIPOS DE PRUEBAS

- ♦ Pruebas estáticas.
- ♦ Pruebas dinámicas.

LOS PRINCIPIOS DE LA PRUEBA DE SISTEMAS

- 1. La prueba es el proceso de ejecutar un programa con la intención de encontrar errores** (parece un enfoque destructivo. Actitud correcta: una prueba exitosa es aquella que encuentra un error).
- 2. Es imposible probar completamente cualquier módulo no trivial o cualquier sistema.**
- 3. La prueba implica creatividad y trabajo duro.**
- 4. La prueba puede prevenir posibles errores,** cuando se realizan a través de las diferentes etapas del ciclo de vida.
- 5. Es mejor que las pruebas sean realizadas por personas diferentes a quienes hicieron el desarrollo del sistema.**

PROCEDIMIENTOS Y TÉCNICAS GENERALES DE PRUEBA

1. Use siempre datos de entrada bien definidos para los que se conozcan los resultados correctos que deben obtenerse.
2. Detecte primero los defectos obvios (usando datos de prueba muy simples) y luego sí realice pruebas más complejas.
3. Cuando modifique algo mientras prueba realice un solo cambio cada vez y utilice los mismos datos con los que detectó el defecto.
4. Pruebe el programa para verificar si detecta entradas incorrectas.

PRUEBAS DE INTEGRACIÓN

- Pruebas de estructura
- Pruebas funcionales (de los subsistemas)
- Pruebas de desempeño y de esfuerzo

ETAPAS INVOLUCRADAS EN TODAS LAS PRUEBAS

- 1. Seleccionar qué es lo que debe medir la prueba,** es decir, cuál es su objetivo, para qué exactamente se hace la prueba.
- 2. Decidir cómo se va a realizar la prueba,** es decir, qué clase de prueba se va a utilizar para medir la calidad escogida y qué clase de elementos de prueba se deben usar.
- 3. Desarrollar los casos de prueba. Un caso de prueba** es un conjunto de datos o situaciones de prueba que se utilizarán para ejecutar la unidad que se prueba o para revelar algo sobre el atributo de calidad que se está midiendo.
- 4. Determinar cuáles deberían ser los resultados esperados o correctos de los casos de prueba** y crear el documento con los casos y sus resultados esperados, denominado **oráculo de prueba**, antes de realizar la prueba.
- 5. Ejecutar los casos de prueba.**
- 6. Comparar los resultados de la prueba con los resultados esperados.** Cualquier discrepancia entre ellos significa un error. Típicamente el error está en el sistema o unidad probada, pero también

puede ser generado por algún aspecto del mismo proceso de prueba o en el oráculo de prueba.