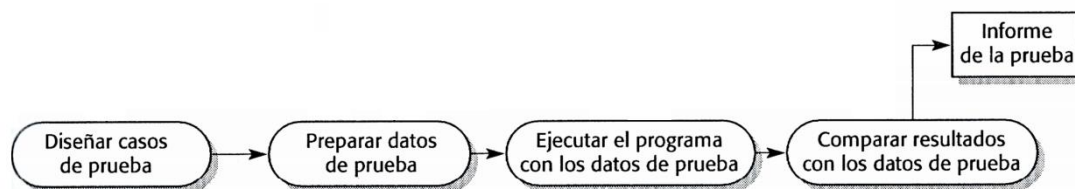


## TEMA 7. PRUEBAS DE SOFTWARE

### 1. INTRODUCCIÓN

Las pruebas software solo pueden demostrar la presencia de errores, no su ausencia. El objetivo es convencer de que el software es lo suficientemente bueno, para lo cual, se realizan **pruebas de validación**. El proceso de pruebas se realiza de la siguiente forma:



Los **programadores** se encargan de comprobar los componentes que ellos han desarrollado y el **equipo de integración** prueba el sistema con todos los componentes.

### 2. PRUEBAS DE COMPONENTES

Las realizan los programadores del componente y consisten en pruebas de defectos que incluyen:

- **Pruebas de métodos o funciones** con diferentes parámetros de entrada.
- **Pruebas de clases** donde se prueban todas las operaciones y todos los estados por los que puede pasar. En caso de herencia se deben probar también las operaciones heredadas o sobrescritas por algún hijo.

Cuando un componente está formado por varios objetos que interactúan, se hacen **pruebas de interfaz**. Los errores comunes suelen ser un mal uso de la interfaz o no comprender bien el funcionamiento del componente.

Principios y Fundamentos de la Ingeniería de Software

### 3. PRUEBAS DEL SISTEMA

Las realiza el equipo de integración para probar cómo trabaja el sistema al unir los componentes. Las pruebas de sistema tienen dos fases:

- **Pruebas de integración:** se accede al código fuente para buscar errores.
- **Pruebas de entrega:** se comprueba si el sistema satisface los requisitos. Si participa el cliente son **pruebas de aceptación**.

#### Pruebas de integración

Los componentes pueden ser: comerciales, reutilizables o nuevos desarrollados. En los sistemas complejos pueden estar los tres tipos. Para facilitar la localización de errores, se hacen las pruebas de forma **incremental** de menos a más, empezando por una configuración mínima.

A la hora de elegir el componente para añadir en el incremento, pueden darse dos casos:

- Lo decide el cliente (metodología ágil).
- Lo decide el equipo empezando por los que se usan con más frecuencia.

En las pruebas incrementales, cuando se añade un componente, se vuelven a ejecutar las pruebas previas por si al añadir el nuevo componente se detectan errores. Es lo que se conoce como **pruebas de regresión**.

#### Pruebas de entrega

Son pruebas de **caja negra** (pruebas funcionales) donde se verifica que cumple con los requisitos, no falla durante su uso normal y funciona a buen rendimiento. Lo que se intenta es hacer que el sistema falle forzando entradas y salidas inválidas.

### 4. DISEÑO DE CASOS DE PRUEBA

Los pasos son:

1. Seleccionar característica a probar.
2. Seleccionar datos de entrada.
3. Documentar las salidas esperadas.
4. Diseñar prueba automatizada (opcional).

Las técnicas que se utilizan para diseñar los casos de pruebas son:

- **Pruebas basadas en requisitos:** comprueba que el sistema cumple con los requisitos.
- **Pruebas de particiones:** las particiones son grupos de datos con características comunes. El objetivo es que el sistema ejecute las entradas de todas las particiones y genere salidas en todas las particiones.
- **Pruebas estructurales:** se ejecutan todas las sentencias del código al menos una vez. Son llamadas pruebas de **caja blanca**.

Raúl Castilla Bravo



- ✓ ESCUELAS 40 ALREDEDOR DEL MUNDO
- ✓ TODOS LOS NIVELES Y OBJETIVOS
- ✓ AÑOS DE 80 EXPERIENCIA
- ✓ 97% DE RECOMENDACIÓN