Ingeniería de Software II

Estrategias de prueba PARTE 1

2025

Enfoque estratégico de pruebas

Una estrategia de pruebas del software proporciona una guía que describe los pasos a seguir, cuándo se planean y llevan a cabo, cuánto esfuerzo, tiempo y recurso se requerirán.

Enfoque estratégico de pruebas

- La prueba es un **conjunto de actividades** que se planean con anticipación y se realizan de manera sistemática.
- Conjunto de pasos en el que se incluyen técnicas y métodos específicos del diseño de casos de prueba.
- Una estrategia de pruebas debe incluir pruebas de **bajo nivel y de alto nivel**
- Las actividades de las estrategias de pruebas son parte de la **Verificación y Validación** incluidas en el aseguramiento de la calidad del software

Fuente:

2025

Concepto de Verificación &Validación

»La ***verificación*** es el conjunto de actividades que asegura que el software implemente correctamente una función específica y ***validación*** es un conjunto diferente de actividades que aseguran que el software construido corresponde con los requisitos del cliente.

»**Verificación** : *¿Estamos construyendo el producto correctamente?*

Comprobar que el software está de acuerdo con su especificación, donde se debe comprobar que satisface tanto los requerimientos funcionales como los no funcionales.

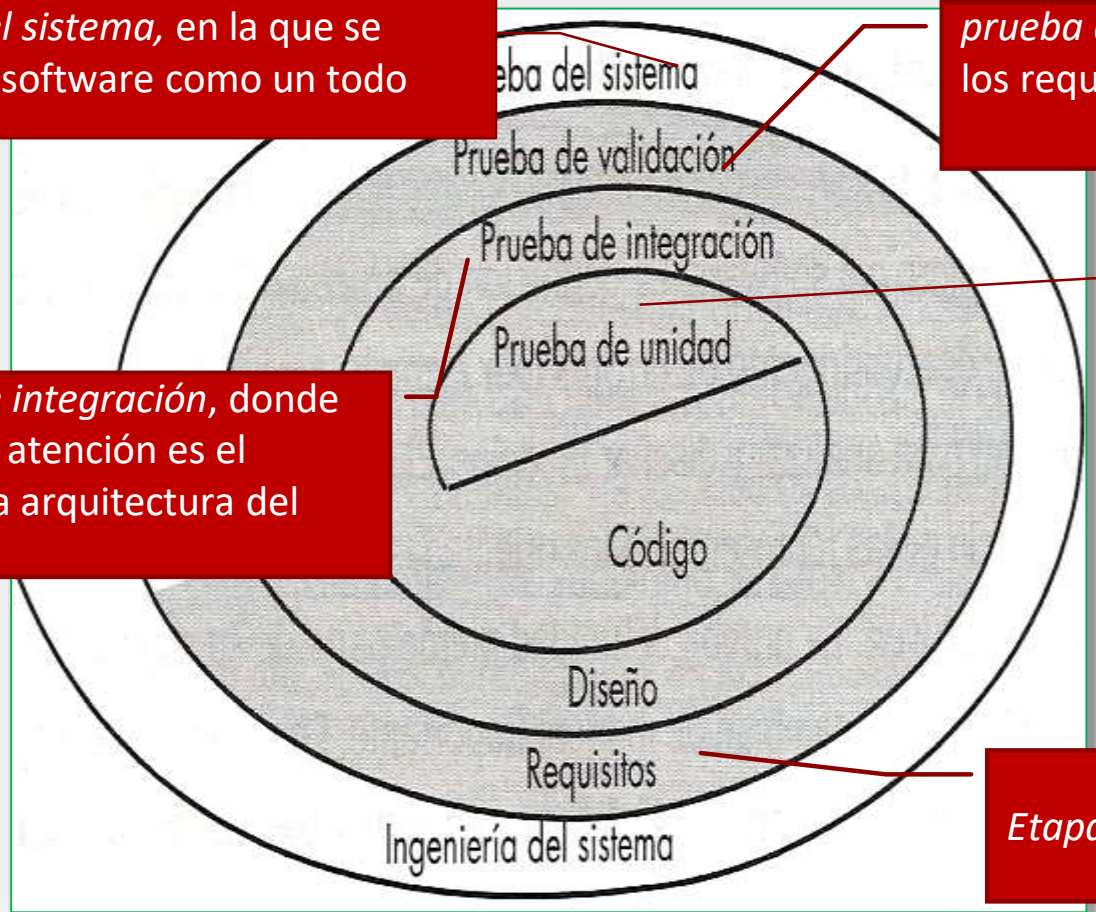
»**Validación** : *¿Estamos construyendo el producto correcto?*

Es un proceso más general, cuyo objetivo es asegurar que el software satisface las expectativas del cliente.

Estrategias de pruebas

prueba del sistema, en la que se prueba el software como un todo

prueba de validación, donde se validan los requisitos establecidos



prueba de unidad , donde comienza la espiral.

prueba de integración, donde el foco de atención es el diseño y la arquitectura del software.

Etapas del proceso de desarrollo

Fuente:

2025

Tipos de Pruebas Software convencionales

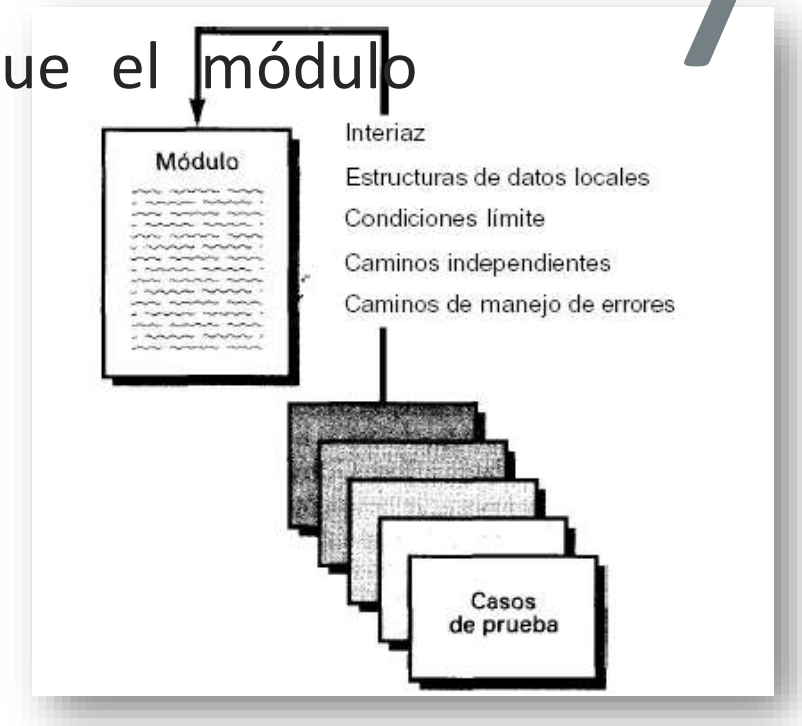
Tipos de Pruebas.
Pruebas de Unidad

»Se prueba la interfaz del módulo para asegurar que la información fluye de forma adecuada.

» Se examinan las estructuras de datos locales.

»Se prueban las condiciones límite para asegurar que el módulo funciona correctamente

»Se ejercitan todos los caminos independientes.



Fuente:

Tipos de Pruebas.
Pruebas de Unidad

- Los errores más comunes detectados por la pruebas de unidad:
  - Cálculos incorrectos
    - *Aplicación incorrecta de predecesores aritméticos*
    - *Operaciones mezcladas*
    - *Inicialización incorrecta*
    - *Falta de precisión*
    - *Representación simbólica incorrecta*
  - Comparaciones erróneas
  - Flujos de control inapropiados

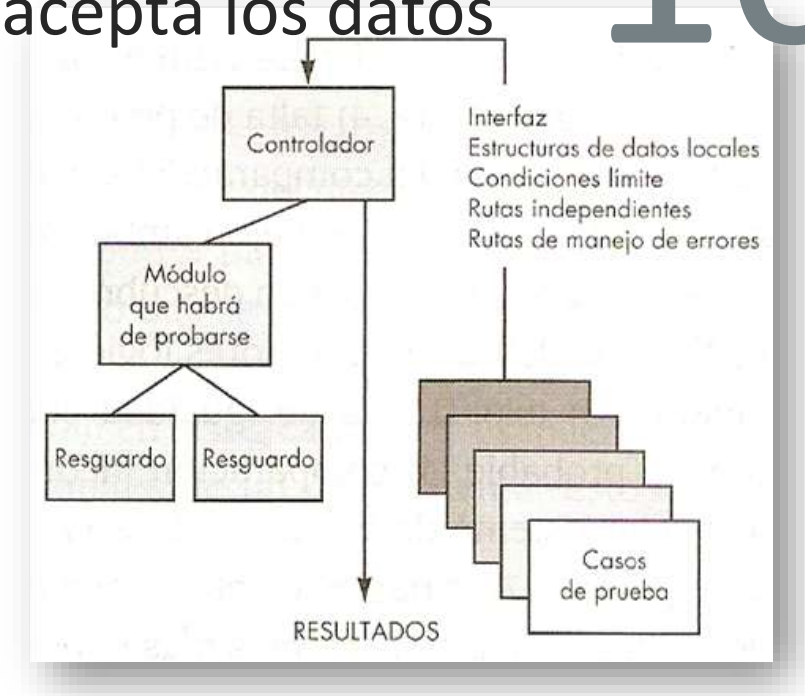
Tipos de Pruebas.
Pruebas de Unidad

- Los Casos de prueba deben descubrir errores como:
  - Comparaciones entre diferentes tipos de datos
  - Operadores lógicos aplicados incorrectamente
  - Expectativas de igualdad con grado de precisión
  - Comparación incorrecta de variables
  - Terminación inapropiada o inexistente de bucles
  - Falla en la salida cuando se encuentre una iteración divergente
  - Variables de bucle modificadas inapropiadamente

Tipos de Pruebas.
Pruebas de Unidad - Procedimiento

- Como un componente no es un programa independiente, se debe desarrollar para cada prueba de unidad un software que controle y/o resguarde.
- Un controlador es un «programa principal» que acepta los datos del caso de prueba, pasa estos datos al módulo

  (a ser probado) y muestra los resultados.



Fuente: Pressman Cap. 18

Tipos de Pruebas.
Pruebas de Unidad - Procedimiento

- Un resguardo sirve para reemplazar a módulos subordinados al componente que hay que probar.
- Los controladores y resguardos son una sobrecarga de trabajo.
- Si los controladores y resguardos son sencillos, el trabajo adicional es relativamente pequeño.
- La prueba de unidad se simplifica cuando se diseña un módulo con un alto grado de cohesión.

Fuente:

2025

Tipos de Pruebas.
Pruebas de Integración

- Se toman los componentes que han pasado las pruebas de unidad y se los combina según el diseño establecido.
- En esta combinación es posible que:
  - Los datos se pueden perder en una interfaz.
  - Un módulo puede tener un efecto adverso e inadvertido sobre otro.
  - La combinación de subfunciones no produzca el resultado esperado.

Tipos de Pruebas.
Pruebas de Integración

- El programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y de corregir
- La Integración puede ser :
  - Descendente
  - Ascendente

Tipos de Pruebas.
Pruebas de Integración - *Descendente*

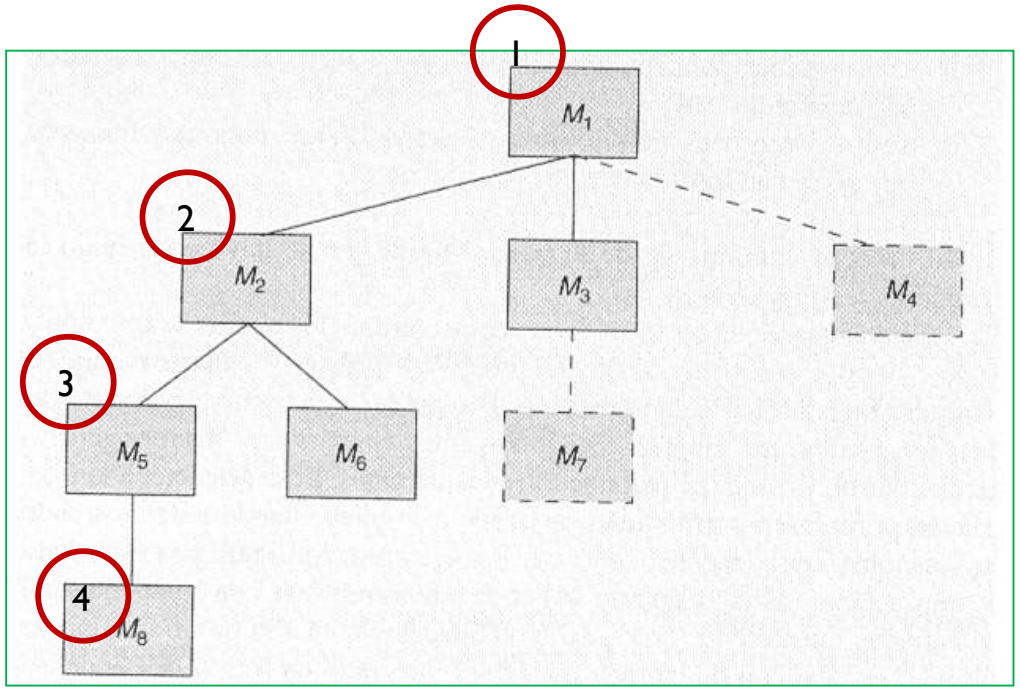
- Los módulos se integran al descender por la jerarquía de control, iniciando por el programa principal
- Se puede realizar:
  - En profundidad
    - *Primero-en-profundidad integra todos los módulos de un camino de control principal de la estructura.*
  - En anchura
    - *Primero-en-anchura incorpora todos los módulos directamente subordinados a cada nivel*

Tipos de Pruebas.
Pruebas de Integración - *Descendente*

»En profundidad: primero-en-profundidad integra todos los módulos de un camino de control principal de la estructura.

Pasos:

1. Conductor: Módulo principal
   Resguardos: Para los módulos subordinados
2. Sustituir resguardos por módulos 1 a 1
3. Probar
4. Reemplazar otro resguardo
5. Pruebas de regresión



Fuente:

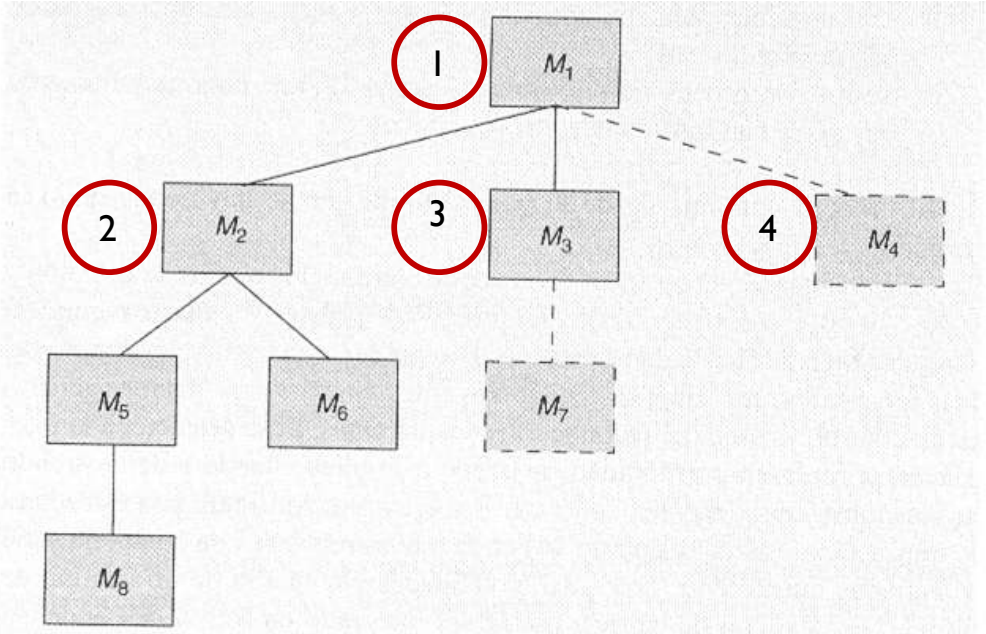
Tipos de Pruebas.
Pruebas de Integración - *Descendente*

»**<u>En anchura</u>**: primero-en-anchura incorpora todos los módulos directamente subordinados a cada nivel.

Pasos:

1. Conductor: Módulo principal

   Resguardos: Para los módulos subordinados
2. Sustituir resguardos por módulos 1 a 1
3. Probar
4. Reemplazar otro resguardo
5. Pruebas de regresión



Fuente:

Tipos de Pruebas.
Pruebas de Integración - *Ascendente*

- Se empieza la prueba con los módulos atómicos (es decir, módulos de los niveles más bajos de la estructura del programa).

- Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos pero no así, los conductores.

Tipos de Pruebas.
Pruebas de Integración - *Ascendente*

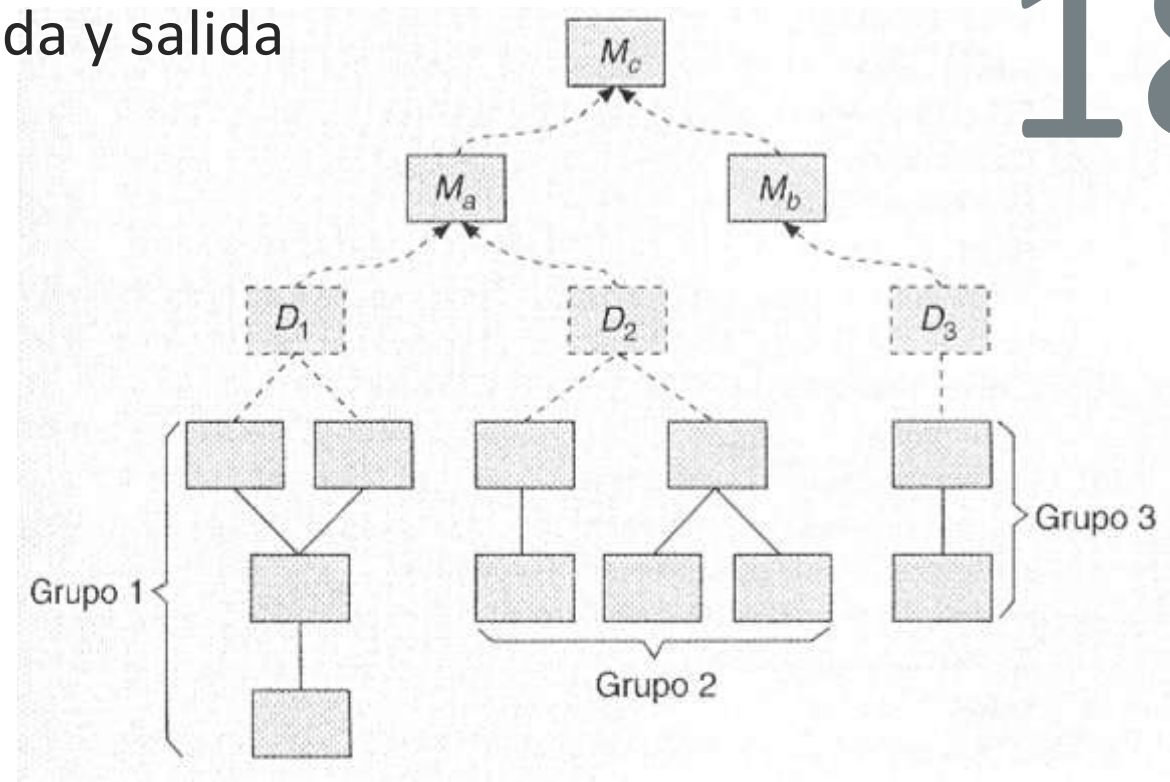
Pasos :

»1.Combinar módulos de bajo nivel

»2.Hacer conductor para coordinar entrada y salida

»3. Probar el grupo

»4.Eliminar conductores



Fuente:

Tipos de Pruebas.
Pruebas de integración - *Selección*

Hay discusión respecto a las ventajas y desventajas de los enfoques Ascendente con Descendente.

La principal desventaja del enfoque Descendente es la necesidad de los resguardos.

La principal desventaja de la opción Ascendente es que “el programa como entidad no existe hasta que se agrega el último módulo”.

La elección depende de las características del software,

Un enfoque combinado (*prueba sándwich*) puede ser el mejor.

Tipos de Pruebas.
Pruebas de integración - *Pruebas de regresión*

- Cada vez que se añade un nuevo módulo como parte de una Prueba de integración, el software cambia.
- Se establecen nuevos caminos, pueden ocurrir nuevas E/S y se invoca una nueva lógica de control.
- Estos cambios pueden causar problemas con funciones que antes trabajaban perfectamente.
- En el contexto de una estrategia de Prueba de integración, la Prueba de regresión es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos colaterales no deseados.

Tipos de Pruebas.
Pruebas de integración - *Pruebas de regresión*

- Estas pruebas se pueden hacer manualmente, volviendo a realizar un subconjunto de todos los casos de prueba o utilizando herramientas automáticas.
- El conjunto de pruebas de regresión contiene tres clases diferentes de casos de prueba:
  - una muestra representativa de pruebas que ejercite todas las funciones del software.
  - pruebas adicionales que se centren en las funciones del software que son probablemente afectadas por el cambio.
  - pruebas que se centren en los componentes del software que han cambiado.

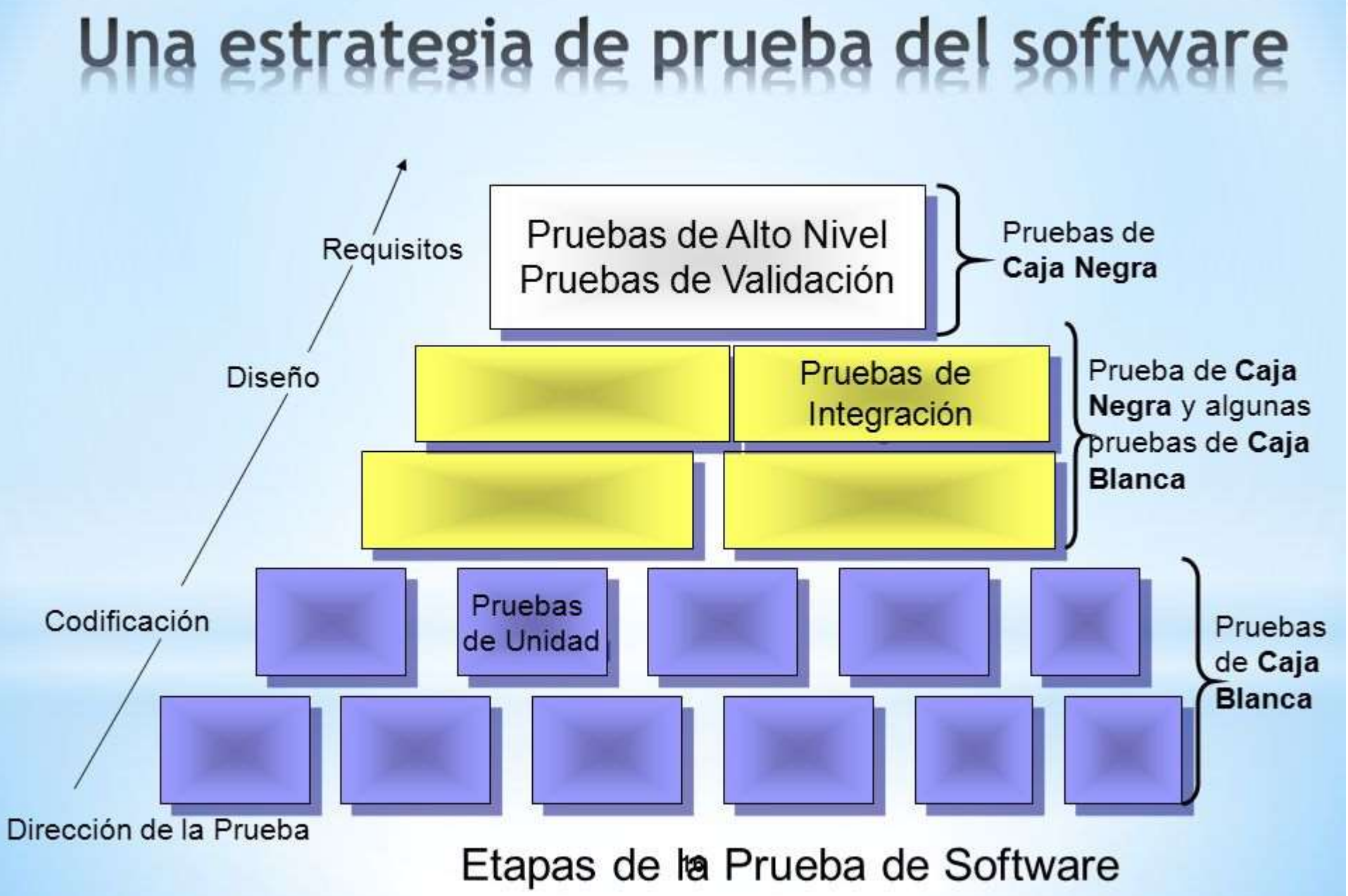
Tipos de Pruebas.
Pruebas de integración - *Criticidad*

Se deben identificar los módulos críticos, que pueden ser los que:

1. *Abordan muchos requerimientos de software*
2. *Tienen alto nivel de control*
3. *Es complejo o proclive a error*
4. *Tiene requerimientos de rendimientos definidos.*

Deben probarse lo antes posible. Las pruebas de regresión deben hacer foco en ellos.

Posible estrategia de pruebas de integración



Tipos de Pruebas.
Pruebas de Unidad e Integración para software OO

»Prueba de Unidad :

Por lo general una clase encapsulada es el foco de la prueba de unidad.

Los métodos son las unidades comprobables más pequeñas.

La prueba de clase es el equivalente en este caso, la cual debe ser dirigida a las operaciones encapsuladas por la clase y el comportamiento de estado de ésta.

»Prueba de integración:

El software OO no tiene una estructura de control jerárquico obvia.

La prueba basada en hebra integra el conjunto de clases requeridas para responder a una entrada o evento.

La prueba basada en uso comienza con las clases independientes, luego las dependientes.

Tipos de Pruebas.
Pruebas del Sistema

- La prueba del sistema, está constituida por una serie de pruebas diferentes.
- Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

Tipos de Pruebas.
Pruebas del Sistema

Tipos de Pruebas.
Pruebas de Validación

- La validación del software se consigue mediante una serie de pruebas que demuestren la conformidad con los requisitos.
- Una vez que se procede con cada caso de prueba de validación, puede darse una de las dos condiciones:

Las características de funcionamiento o de rendimiento están de acuerdo con las especificaciones y son aceptables;

o

Se descubre una desviación de las especificaciones y se crea una lista de deficiencias.

Tipos de Pruebas.
Pruebas de Validación

- Comienzan cuando finalizan las pruebas de integración.

- Revisión de la configuración

  Asegurar que todos los elementos de la configuración del software se hayan desarrollado apropiadamente, estén catalogados y contengan detalle suficiente para reforzar la fase de soporte.

Tipos de Pruebas.
Pruebas de Validación

- Pruebas de **aceptación (ALFA y BETA)**

  Las realiza el usuario final en lugar del responsable del desarrollo del sistema, una prueba de aceptación puede ir desde algo informal, hasta la ejecución sistemática de una serie de pruebas bien planificadas.

  Dentro de las Pruebas de aceptación se pueden encontrar:

  Pruebas ALFA: desarrolladores con clientes antes de liberar el producto.

  Pruebas BETA: seleccionando los clientes que efectuarán la prueba. El desarrollador no se encuentra presente.

Tipos de Pruebas.
Pruebas de Validación – aceptación ALFA

- Se llevan a cabo, por un cliente, en el lugar de desarrollo.
- Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso.
- Las pruebas alfa se hacen en un entorno controlado.
- Se realizan después de que todos los procedimientos de prueba básicos, como las pruebas unitarias y pruebas de integración se han completado, y se produce después de las pruebas del sistema.
- Esta no es la versión final de software y cierta funcionalidad puede ser añadido al software incluso después de la prueba alfa.

Fuente:

Tipos de Pruebas.
Pruebas de Validación – aceptación BETA

- Se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes.
- El desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador.
- El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.
- Las pruebas beta es la última fase de las fases de prueba y se hace utilizando técnicas de caja negra.
- A veces la versión beta también es liberada en el mercado, y en base a las modificaciones que se hacen comentarios de los usuarios o si no hay cambios en el software se libera.

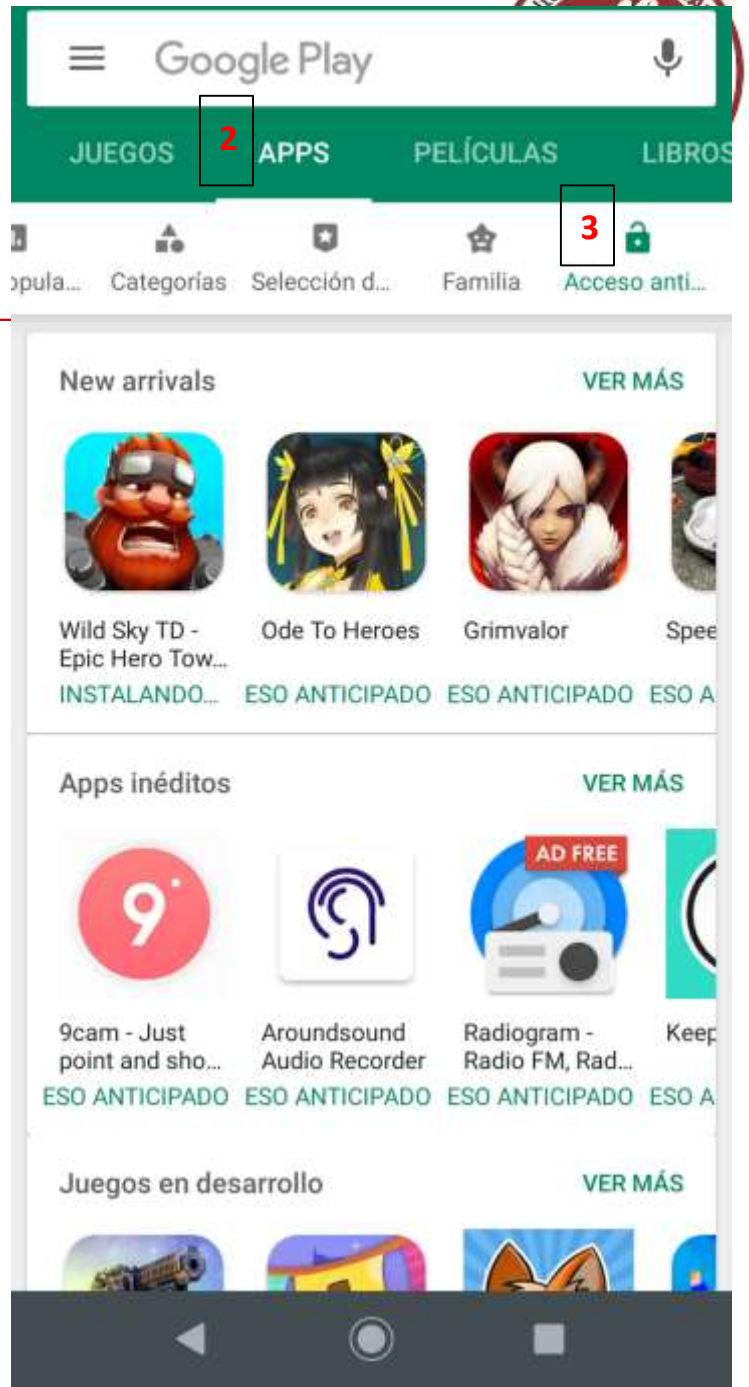
Fuente:

Tipos de Pruebas.
Pruebas BETA

Beta testers de App de Android

¿Cómo hacer?

1. Ir a Play Store
2. Buscar pestaña de Apps
3. Buscar opción Acceso Anticipado, (está al final)
4. Descargar la app que se quiera probar
5. Se habilitan mensajes privados con el desarrollador



Fuente:

Tipos de Pruebas.
Pruebas de Validación – aceptación BETA

»Ejemplo de prueba beta:

https://developer.apple.com/testflight/

»La explicación de cómo desarrollar una prueba beta de pre lanzamiento de una App para Apple Store se puede ver en:

https://developer.apple.com/library/content/documentation/LanguagesUtilities/Conceptual/iTunesConnect_Guide_ES/Chapters/BetaTestingTheApp.html

Tipos de Pruebas.
Pruebas de Validación – aceptación BETA

»Lugares donde pueden ser beta tester:

https://www.usertesting.com/

https://www.utest.com/

Beta Testers: Únete a la comunidad de BetaTesting hoy

Una estrategia possible de pruebas

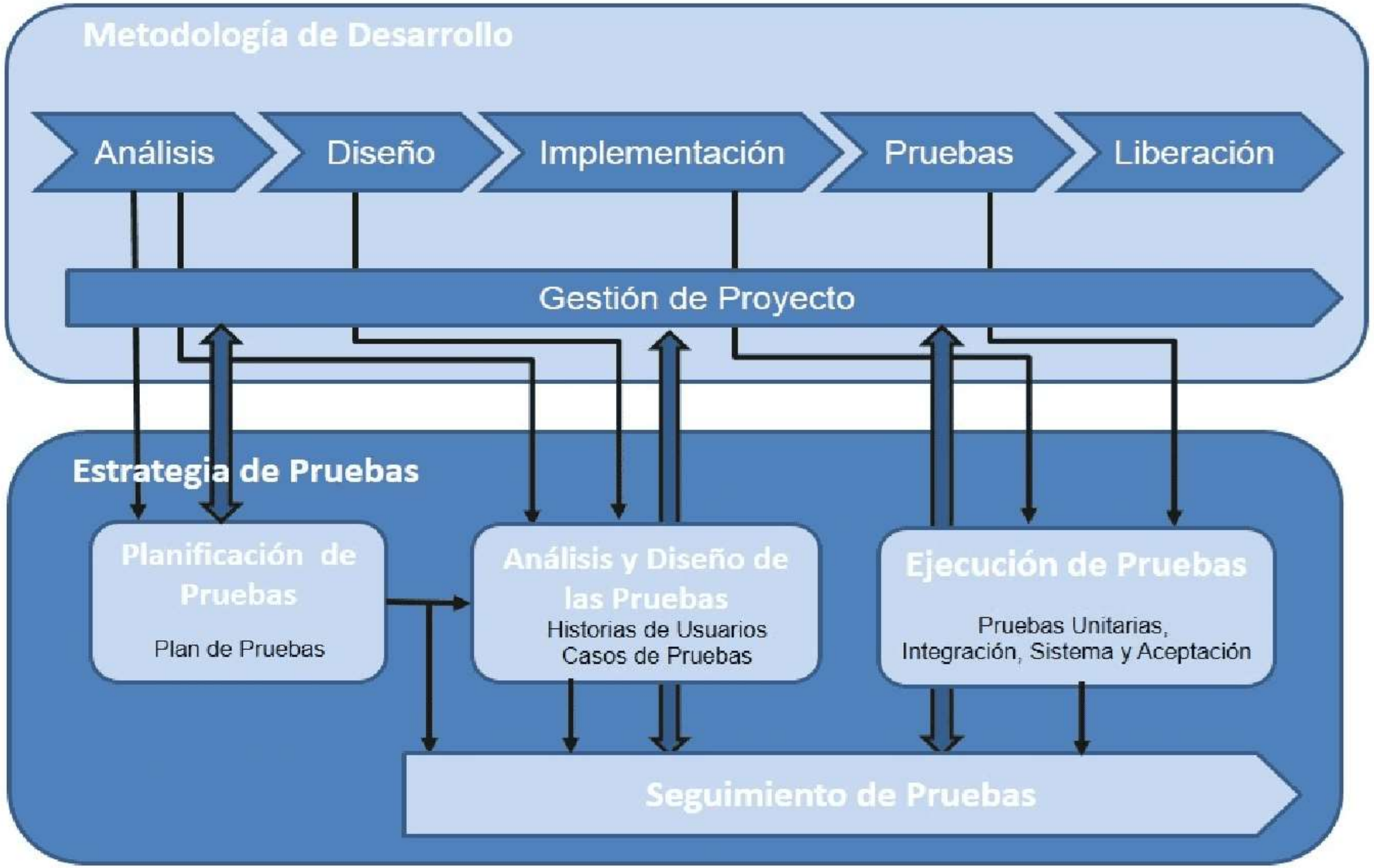


Imagen tomada de https://www.gestiopolis.com/sistema-gestion-pruebas-soluciones-erp-enfocadas-paquetes-odoo-cuba/

Prueba de entornos especializados

» A medida que el software se hace más complejo, crece también la necesidad de enfoques de pruebas especializados.

»Pruebas de interfaces gráficas

»Pruebas de arquitecturas cliente-servidor

»Pruebas de la documentación y ayuda

»Pruebas de sistema en tiempo real

Fuente:

Prueba de entornos especializados
Prueba de arquitectura cliente-servidor

- Pruebas de funcionalidad de la aplicación
- Prueba de servidor

  Probar las funciones de coordinación y manejo de datos del servidor.

  Desempeño del servidor (tiempo de respuesta y procesamiento total de los datos)
- Prueba de base de datos

  Probar la exactitud e integridad de los datos, examinar transacciones, asegurar que se almacenan, actualizan y recuperan los datos.
- Pruebas de transacciones

  Se crea una serie de pruebas para asegurar que cada transacción se procese de acuerdo a los requisitos.
- Pruebas de comunicación de red

  Verificar comunicación entre los nodos, que el paso de mensajes, transacciones y tráfico de la red se realice sin errores.

Fuente:

Prueba de entornos especializados
Prueba de la documentación y funciones de ayuda

- Es importante para la aceptación del programa.
- Revisar la guía del usuario o funciones de ayuda en línea.
- Prueba de documentación es en dos fases:

  Revisar e inspeccionar

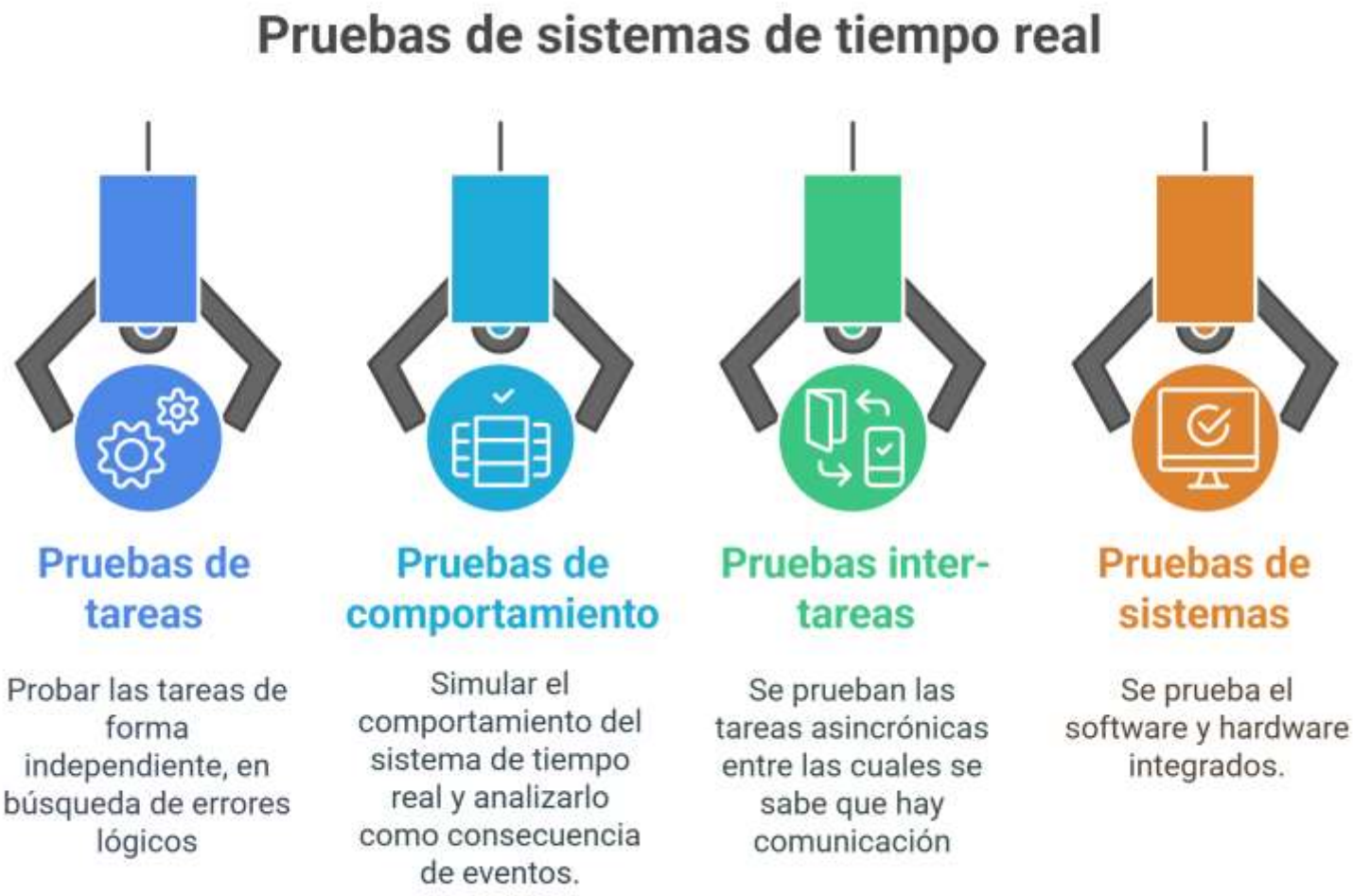
  examinar la claridad editorial del documento.

  Prueba en vivo

  usar la documentación junto con el programa real.

Pruebas de sistemas de tiempo real

El diseño de los casos de prueba, además de los convencionales deben incluir manejo de eventos (interrupciones), temporización de los datos, el paralelismo entre las tareas, etc.



DEPURACIÓN de Programas

2025

Depuración

La depuración de programas, es el proceso de identificar y corregir errores en programas informáticos.

La depuración no es una prueba, pero siempre ocurre como consecuencia de la prueba efectiva.

Es decir, se descubre un error, la depuración elimina dicho error.

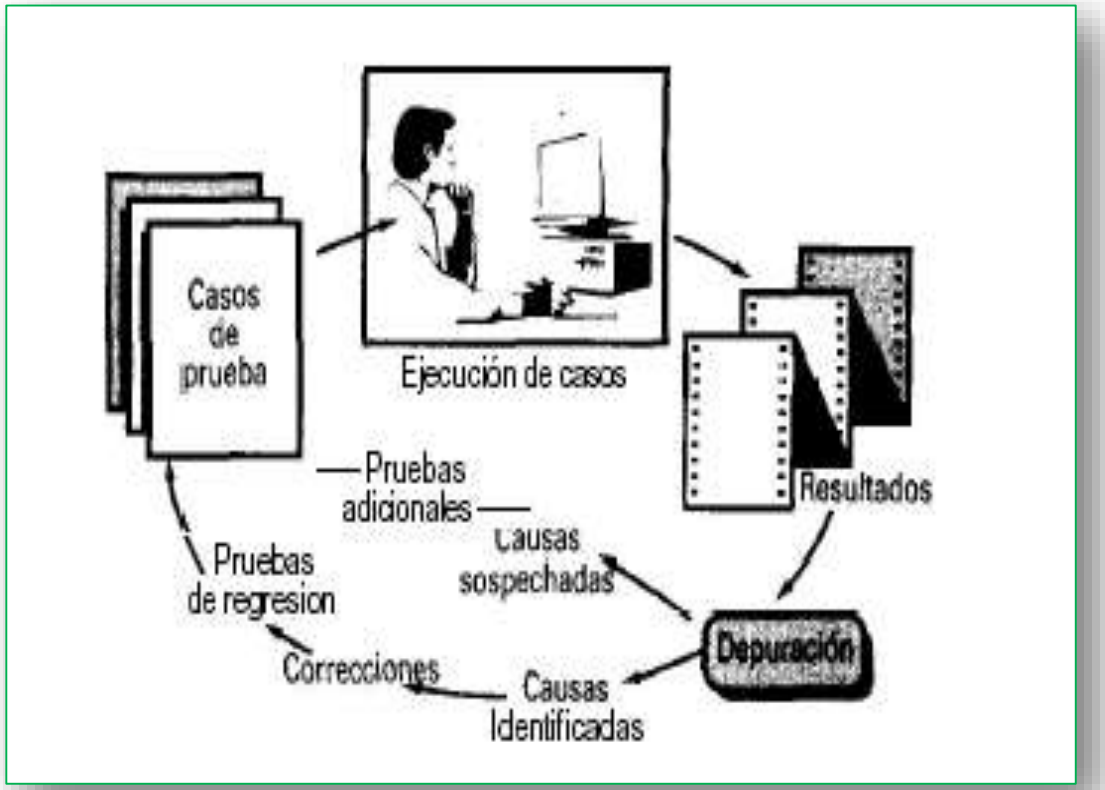
El Proceso de Depuración

»El proceso de depuración siempre tiene uno de los dos resultados:

Se encuentra la causa, se corrige y se elimina.

o

No se encuentra la causa. La persona que realiza la depuración debe sospechar la causa, diseñar un caso de prueba que ayude a confirmar sus sospechas y el trabajo vuelve hacia atrás a la corrección del error de una forma iterativa.



Fuente:

El Proceso de Depuración
Características de los errores

1. Síntoma lejano (geográficamente) de la causa
2. Síntoma desaparece temporalmente al corregir otro error
3. Síntoma producido por error
4. Síntoma causado por error humano
5. Síntoma causado por problemas de tiempo
6. Condiciones de entrada difíciles de reproducir
7. Síntoma intermitente (especialmente en desarrollos hardware-software)
8. El síntoma se debe a causas distribuidas entre varias tareas que se ejecutan en diferentes procesadores

Fuente:

Enfoques de la Depuración