

ANALISIS Y DISEÑO DE SISTEMAS – UML PUDS.

VALIDACION DEL DISEÑO – PRUEBAS UNITARIAS Y FUNCIONALES DEL DISEÑO.

Introducción

- **Contenido:**

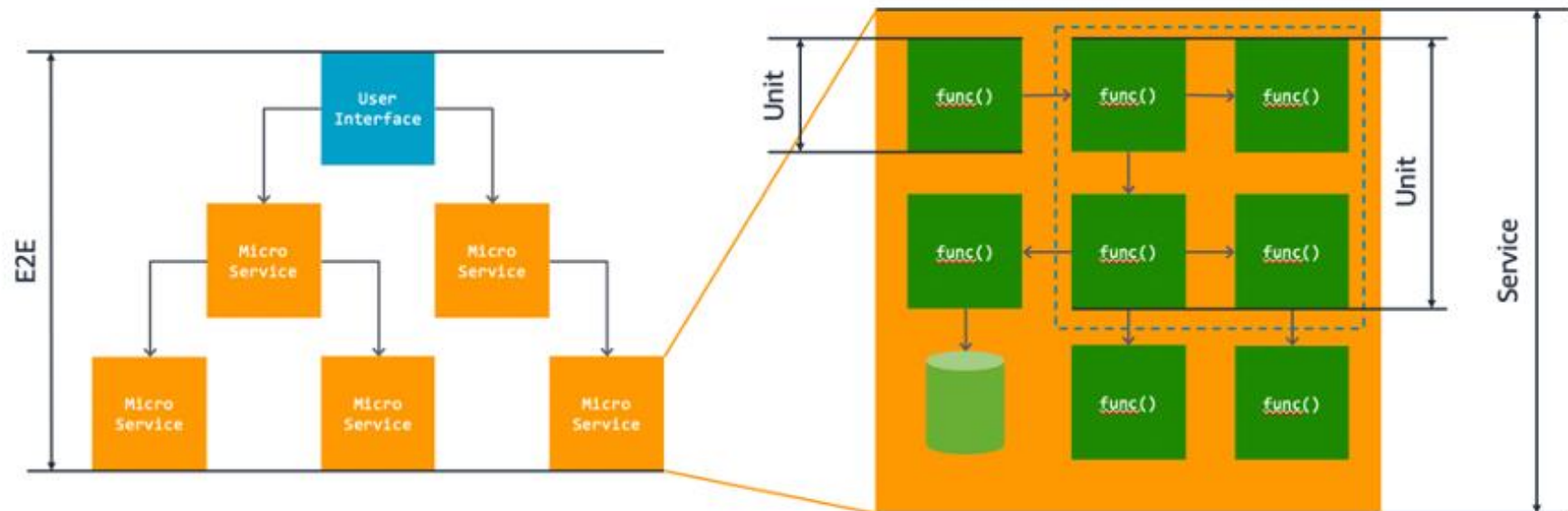
- Importancia de las pruebas en el desarrollo de software.
- Diferencia entre pruebas unitarias y funcionales.
- Objetivo de la presentación: Entender y aplicar ambos tipos de pruebas.

¿Qué son las Pruebas Funcionales?

- Procedimiento para confirmar si el software coincide con su diseño.
- Se centra en todo el sistema.
- Valida características, capacidades e interacción del sistema.

¿Qué son las Pruebas Unitarias?

- Verificación de unidades y componentes individuales de código.
- Realizadas por desarrolladores al principio del proceso de desarrollo.
- Aislan y prueban pequeñas partes del código.



Diferencias Clave

- **Contenido:**

- Tabla comparativa:
 - Enfoque: Corrección del código vs. Validación del sistema.
 - Nivel de prueba: Código vs. Sistema.
 - Granularidad: Fina vs. Gruesa.
 - Realizadas por: Desarrolladores vs. Probadores dedicados.
 - Cobertura: Pequeña porción vs. Gran parte del sistema.

Pruebas funcionales

1. Unitarias
2. Componentes
3. Integración
4. Sistema
5. Humo
6. Aceptación
7. Regresión
8. De Interfaz

Pruebas no funcionales

1. Rendimiento
2. Carga y Estrés
3. Seguridad
4. Escalabilidad
5. Confiabilidad
6. Instalación
7. Conformidad
8. Compatibilidad

Propósito de las Pruebas

- **Contenido:**

- **Pruebas Unitarias:**

- Aislar y probar partes pequeñas del código.
 - Crear una base de código sólida.
 - Proporcionar documentación para pruebas de alto nivel.

- **Pruebas Funcionales:**

- Probar la funcionalidad de todo el software.
 - Validar la interacción del sistema con la infraestructura.



Mejora de la Calidad del Software

- **Pruebas Unitarias:**

- Ofrecen una red de seguridad para refactorizar el código.
- Obligan a escribir código modular y mantenible.
- Previenen la regresión.

- **Pruebas Funcionales:**

- Aseguran que el sistema funcione según lo previsto.
- Validan la aplicación en el contexto de un usuario real.
- Reducen el riesgo de integrar nuevos códigos.

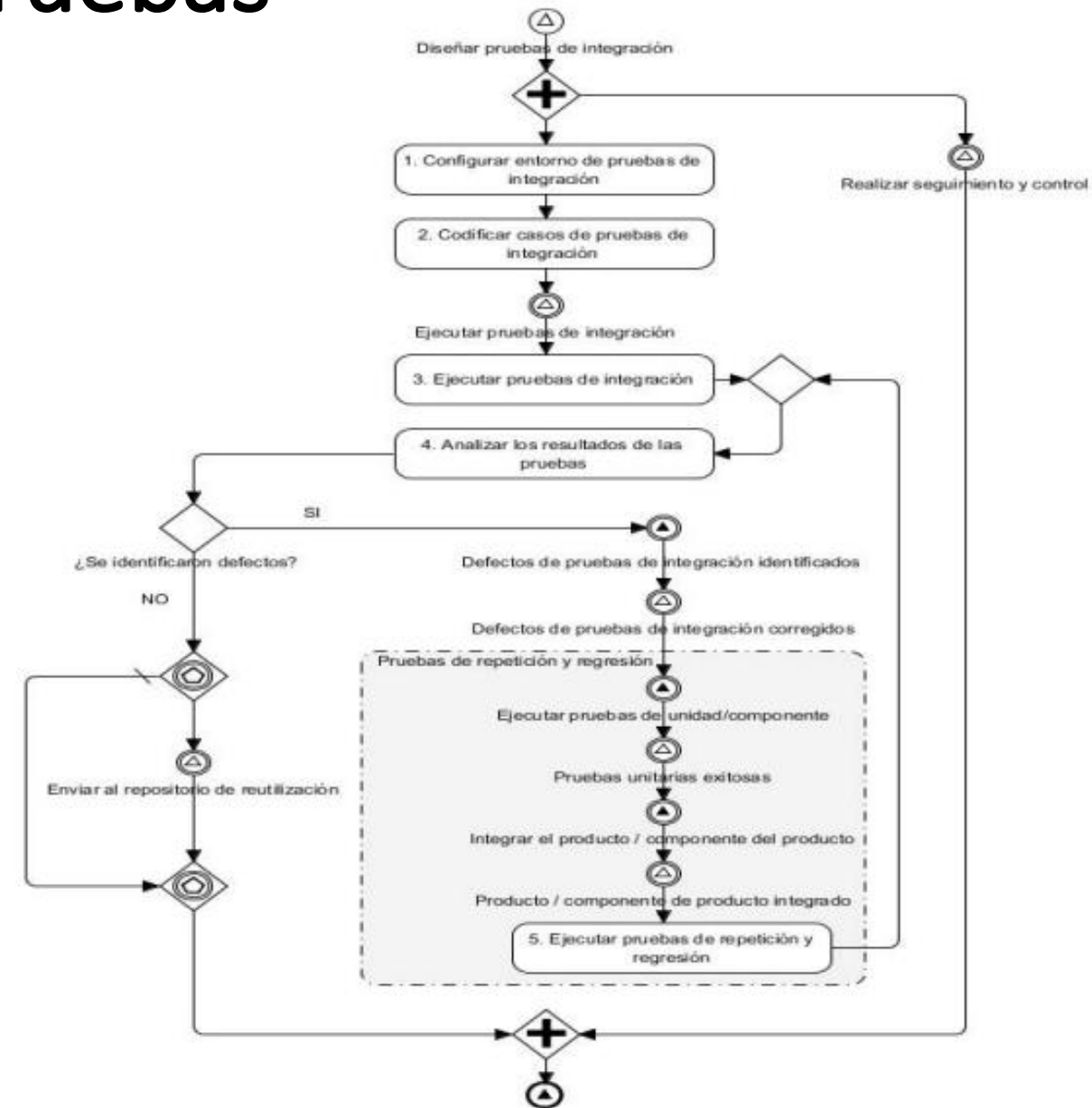
Cuándo Realizar las Pruebas

- **Pruebas Unitarias:**

- Durante la etapa de desarrollo.
- Apoyadas por el desarrollo dirigido por pruebas (TDD).

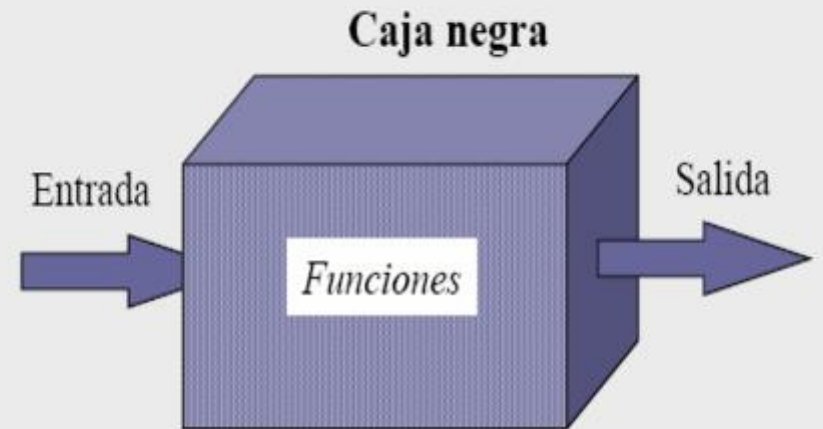
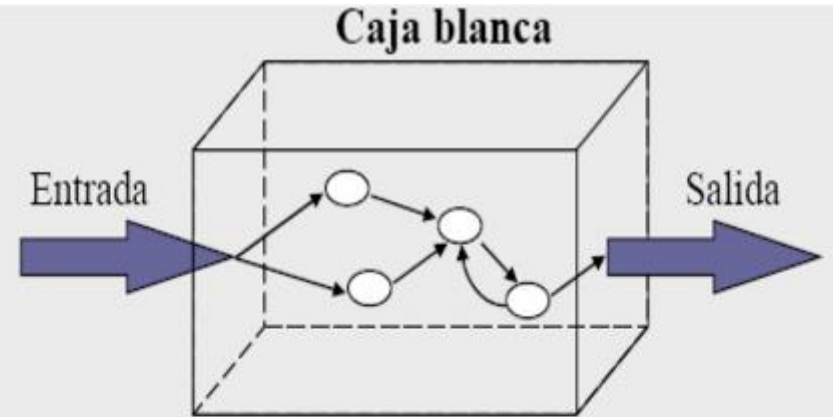
- **Pruebas Funcionales:**

- Cuando dos módulos interactúan.
- En la fase inicial del proyecto.



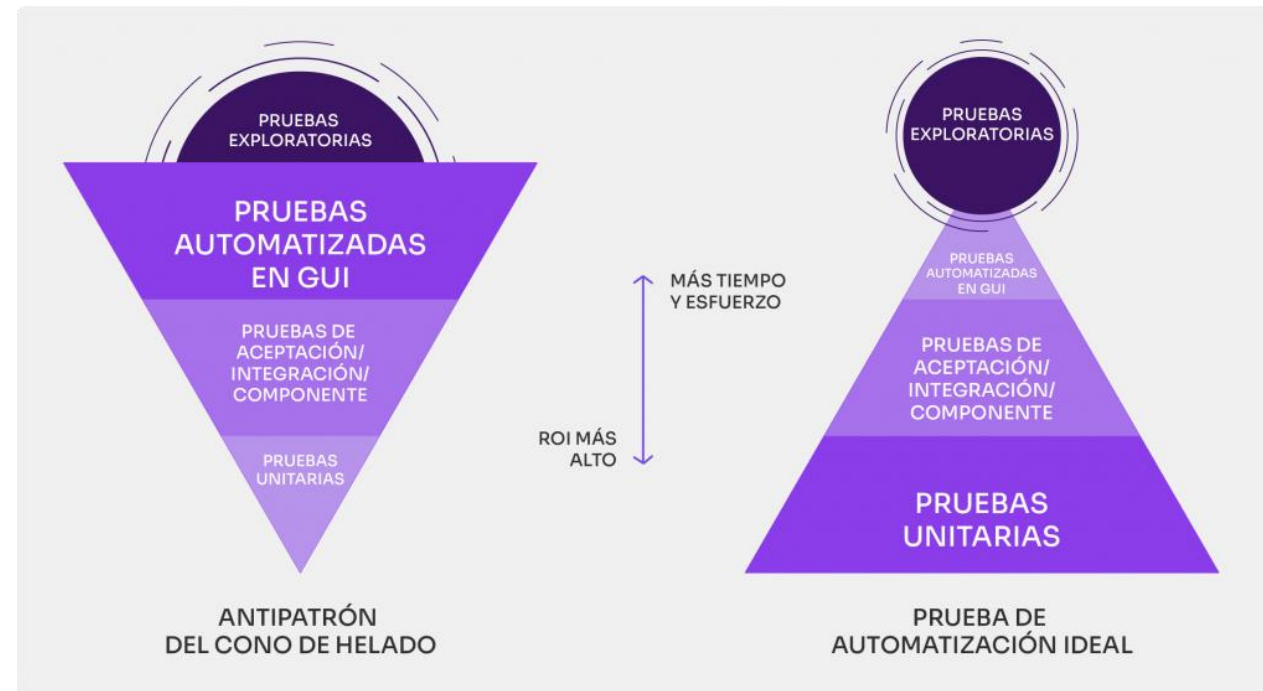
Técnicas de Prueba

- **Pruebas Unitarias:**
 - Técnica de caja blanca.
 - Diseñadas para probar unidades individuales de código.
- **Pruebas Funcionales:**
 - Técnica de caja negra.
 - Prueban la funcionalidad del software sin notar la estructura interna.



Cobertura de Pruebas

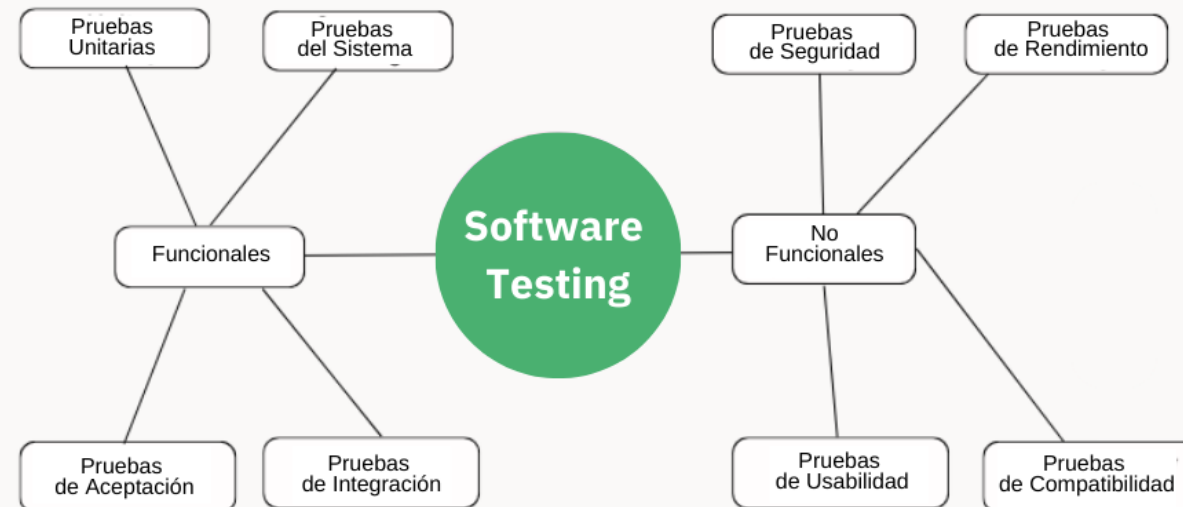
- **Pruebas Unitarias:**
 - Detectan errores antes de producción.
 - Aseguran que se cubran todas las funciones críticas.
- **Pruebas Funcionales:**
 - Establecen la trazabilidad entre requisitos y casos de prueba.
 - Garantizan que se prueben todas las funciones necesarias.



Complejidad al Escribir Casos de Prueba

- **Pruebas Unitarias:**
 - Pueden ser complejas si el diseño del código no está claro.
 - Requieren planificación del diseño del código.
- **Pruebas Funcionales:**
 - Pueden simular todas las dependencias.
 - Requieren un plan de pruebas y mantenimiento.

Clasificación de las Pruebas de Software



Conclusión

- Resumen de la importancia de las pruebas unitarias y funcionales.
- Recomendación de utilizar ambas pruebas de manera complementaria.

Pruebas funcionales

1. Unitarias
2. Componentes
3. Integración
4. Sistema
5. Humo
6. Aceptación
7. Regresión
8. De Interfaz

Pruebas no funcionales

1. Rendimiento
2. Carga y Estrés
3. Seguridad
4. Escalabilidad
5. Confiabilidad
6. Instalación
7. Conformidad
8. Compatibilidad

Fuente:

- <https://www.monografias.com/trabajos94/analisis-y-diseno-sistemas-informacion/analisis-y-diseno-sistemas-informacion>
- https://www.unipamplona.edu.co/unipamplona/portallG/home_109/recursos/octubre2014/administraciondeempresas/semestre7/11092015/analisisydisenosistinformacion.pdf