

# PRUEBAS DE SOFTWARE



Materia: Metodologías de Prueba  
Profesor: Ing. Pablo Andrés Pérez

# DEFINICIÓN: ¿QUE SON LAS PRUEBAS DE SOFTWARE?



- Las pruebas de software ***son un conjunto de actividades diseñadas para evaluar la calidad y el rendimiento de un programa de software.***
- Estas pruebas ***se realizan para verificar que el software funciona como se espera y que cumple con los requisitos especificados.***

# ¿CUANDO REALIZO PRUEBAS?

- Las pruebas de software se pueden realizar **en diferentes etapas del ciclo de vida del desarrollo de software**, desde la fase de diseño hasta la implementación y mantenimiento.
- Se utilizan diferentes técnicas y herramientas de prueba, como pruebas manuales y automatizadas, para identificar errores y defectos en el software.

# OBJETIVO

- *El objetivo principal de las pruebas de software es **mejorar la calidad del software y garantizar que el producto final sea lo más libre de errores posible antes de ser entregado al cliente.***

# METODOLOGIA DE PRUEBAS



- Es importante elegir la metodología de prueba adecuada para el proyecto de software específico, ya que cada una tiene sus fortalezas y debilidades.
- La elección dependerá de ***la naturaleza del software, los requisitos del cliente y el alcance del proyecto.***

# METODOLOGIA DE PRUEBAS

Existen varias metodologías de prueba de software, algunas de las más comunes son:

- **Pruebas manuales:** Se realizan manualmente por los testers, quienes ejecutan los casos de prueba diseñados manualmente. Es una metodología tradicional de prueba de software que implica la interacción manual con el software. Las pruebas manuales son útiles cuando se trata de probar escenarios complejos y no predecibles.
- **Pruebas automatizadas:** Implica el uso de herramientas y programas para automatizar las pruebas. Esta metodología es útil para ejecutar pruebas repetitivas y reducir el tiempo de prueba. Las pruebas automatizadas también son útiles para probar la funcionalidad del software en diferentes entornos y configuraciones.

# METODOLOGIA DE PRUEBAS

- **Pruebas de caja negra:** Se enfoca en la funcionalidad del software sin tener en cuenta el código subyacente. Los testers diseñan casos de prueba basados en los requisitos y las especificaciones de diseño, y los ejecutan para verificar si el software cumple con estos requisitos.
- **Pruebas de caja blanca:** Se enfoca en la estructura interna del software y en cómo funciona el código subyacente. Los testers utilizan herramientas de análisis de código y técnicas de prueba de unidad para identificar problemas en el código.

# METODOLOGIA DE PRUEBAS



- **Pruebas exploratorias:** Esta metodología implica la exploración del software por parte del tester sin un plan o guía específicos. El objetivo es descubrir problemas y errores en el software a través de la interacción con el mismo.
- **Pruebas de seguridad:** Se enfoca en probar la seguridad del software y detectar vulnerabilidades y brechas de seguridad. Los testers utilizan herramientas especializadas y técnicas de prueba de penetración para identificar y explotar las vulnerabilidades del software.



# Definiciones importantes

- PRUEBA (test):

Actividad en la cual se somete a un sistema o uno de sus componentes a una evaluación de los resultados que arroja en base a la ejecución de éste en condiciones especificadas.
- CASO DE PRUEBA (test case):

Conjunto de entradas y condiciones que arrojan resultados esperados desarrollados con un objetivo en particular.

# Definiciones importantes

- ERROR:

Acción humana que produce o genera un resultado incorrecto.

- DEFECTO:

Es la manifestación de un error en el software. Un defecto es encontrado porque causa una FALLA o no, la cuál es una desviación del servicio o resultado esperado.



## Conceptos de Error, Defecto y Fallo

**Error**



Acción humana que produce un resultado incorrecto



**Defecto**



Desperfecto en un componente que puede causar que el mismo falle en sus funciones

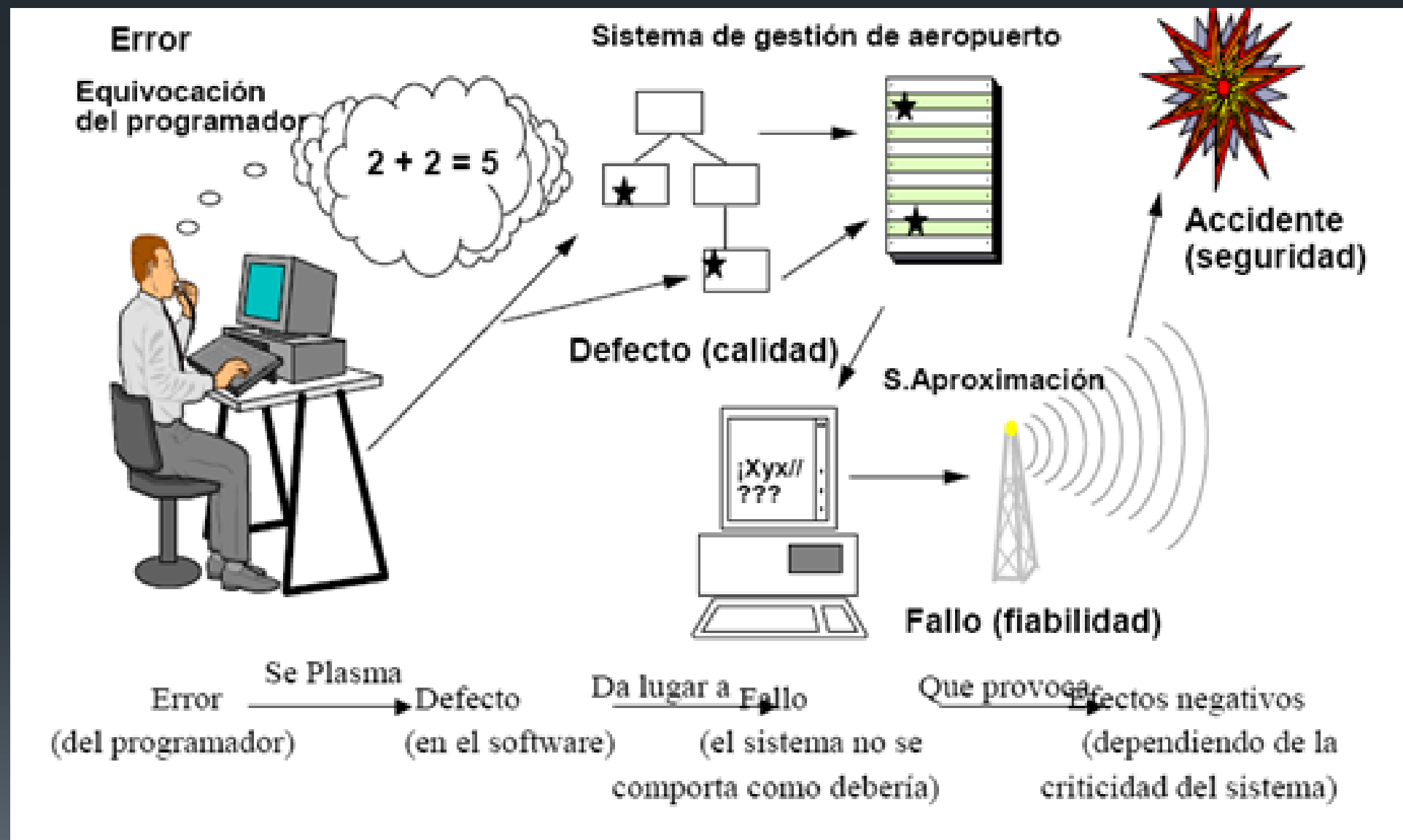


**Fallo**



Manifestación visible de un defecto

# Y entonces...



# DEFINICIONES IMPORTANTES

- VERIFICACIÓN

Determinar si los productos de una fase dada satisfacen las condiciones impuestas al inicio de la fase.

¿Estamos construyendo correctamente el producto?

- VALIDACIÓN:

Evaluación de un sistema o uno de sus componentes durante o al final de su desarrollo para determinar si satisface los requisitos.

¿Estamos construyendo el producto correcto?

# Para que probamos el software?

- Las fallas ocasionan grandes perdidas económicas.
- Son mas costosas encontrarlas y repararlas luego de desarrollar el software.
- Evitar incumplimientos con los clientes (plazos de entrega, presupuestos)
- Para satisfacer al cliente y no perderlo
- Aumentar la productividad y la calidad del producto.
- Reducir el costo en la fase de mantenimiento (automatización de las pruebas)

# Que hay tener en cuenta para hacer pruebas?

- Definir casos de prueba, el máximo posible (¿?)
- Cada caso de prueba debe estar definido para un resultado esperado.
- Los programadores deben evitar probar sus propios programas.
- Revisar a conciencia el resultado de cada prueba y poder identificar posibles síntomas de defectos.
- Los casos de prueba deben incluir datos de entrada validos y esperados como no validos e inesperados

# Que hay tener en cuenta para hacer pruebas?

- Evitar casos no documentados ni diseñados con cuidado.
- No hacer planes de prueba suponiendo que no hay defectos en el sistema (disposición de pocos recursos para las pruebas)



# FUNDAMENTOS DE LAS PRUEBAS

Deben centrarse en 2 fundamentos

- QUE EL SOFTWARE ***HAGA LO QUE DEBE HACER.***
- QUE EL SOFTWARE ***NO HAGA LO QUE NO DEBE HACER*** (efectos secundarios)

# Se puede probar todo???

- No, es imposible evaluar todas las posibilidades.
- Se requieren mas recursos (costo, tiempo y personal)
- ¿Cuántos ciclos de prueba son suficientes?

# ESTÁNDARES INTERNACIONALES

Los siguientes son estándares ampliamente reconocidos y utilizados en la industria en relación a la calidad del software:

- **ISO 9001.**

Norma para la implementación de un método o Sistema de Gestión de la Calidad (SGC), supone la acreditación de la capacidad para satisfacer los requisitos de calidad. Aporta así una serie de requisitos genéricos (no circunscritos al software) y aplicables a cualquier organización.

- **ISO 10005:2018.**

Ofrece una guía para gestionar un plan de calidad todo el ciclo de vida.

- **ISO IEC 25000.**

Supone una familia de normas también conocida como SQuaRE (Software product Quality Requirements and Evaluation), define un marco de referencia para la calidad del producto de software. Evalúa un software en 8 áreas, incluyendo Adecuación funcional, fiabilidad, usabilidad, eficiencia, compatibilidad, seguridad, mantenibilidad y portabilidad.

- **ISO 33000 Calidad de los procesos de desarrollo de software.**

Se enfoca en la evaluación de la calidad de los procesos de desarrollo de software, y también se lo conoce como SPICE (Software Process Improvement and Capability Determination). Así, busca conocer la evolución en el tiempo sobre los procesos de desarrollo, hacer un seguimiento respecto a la competencia y determinar posibles estrategias de mejora.

# ESTÁNDARES INTERNACIONALES

- **CMMI (Capability Maturity Model Integration).**

Proporciona un marco de referencia para evaluar y mejorar la madurez de los procesos en el desarrollo de software. Se centra en áreas como la gestión de proyectos, de la calidad, la configuración, o riesgos, entre otros.

- **ISO 12207 Modelos de Ciclos de Vida del Software.**

Proporciona un estándar para observar los procesos de ciclo de vida del software, desde la idea inicial hasta la retirada del software.

- **IEEE 730 – 2002 Standard for Software Quality Assurance Plans.**

Define qué es el software de alta calidad y propone una elaboración de un Plan de Aseguramiento de la calidad de software (SQAP).

- **ISO 5055 Software Quality Standards.**

En este caso, se trata de medir las debilidades críticas de un software, fijándose en los criterios de seguridad, confianza, eficiencia de rendimiento y mantenibilidad.

# ***ESTÁNDARES INTERNACIONALES***

- ISO 9126 (reemplazada por la 25000)

Modelo de evaluación de calidad de software; aporta características sencillas de evaluación y medición.

- Modelo de calidad
- Métricas externas (Aseguramiento de calidad)
- Métricas internas (Desarrollo)
- Métricas de calidad en uso (Producto final)

# *ESTÁNDARES INTERNACIONALES*

- BS 7925-2
  - Estándar Británico para pruebas de componentes de software (Standard for Software Component Testing)
  - Describe técnicas para el diseño y medición de casos de prueba, trata la ejecución y análisis de los resultados, características a seleccionar para determinar, comparar y mejorar la calidad de la prueba.

# ***ESTÁNDARES INTERNACIONALES***

- IEEE 829

Estándar para documentar pruebas de software. Especifica 8 etapas del proceso de documentación. Planeación de las pruebas

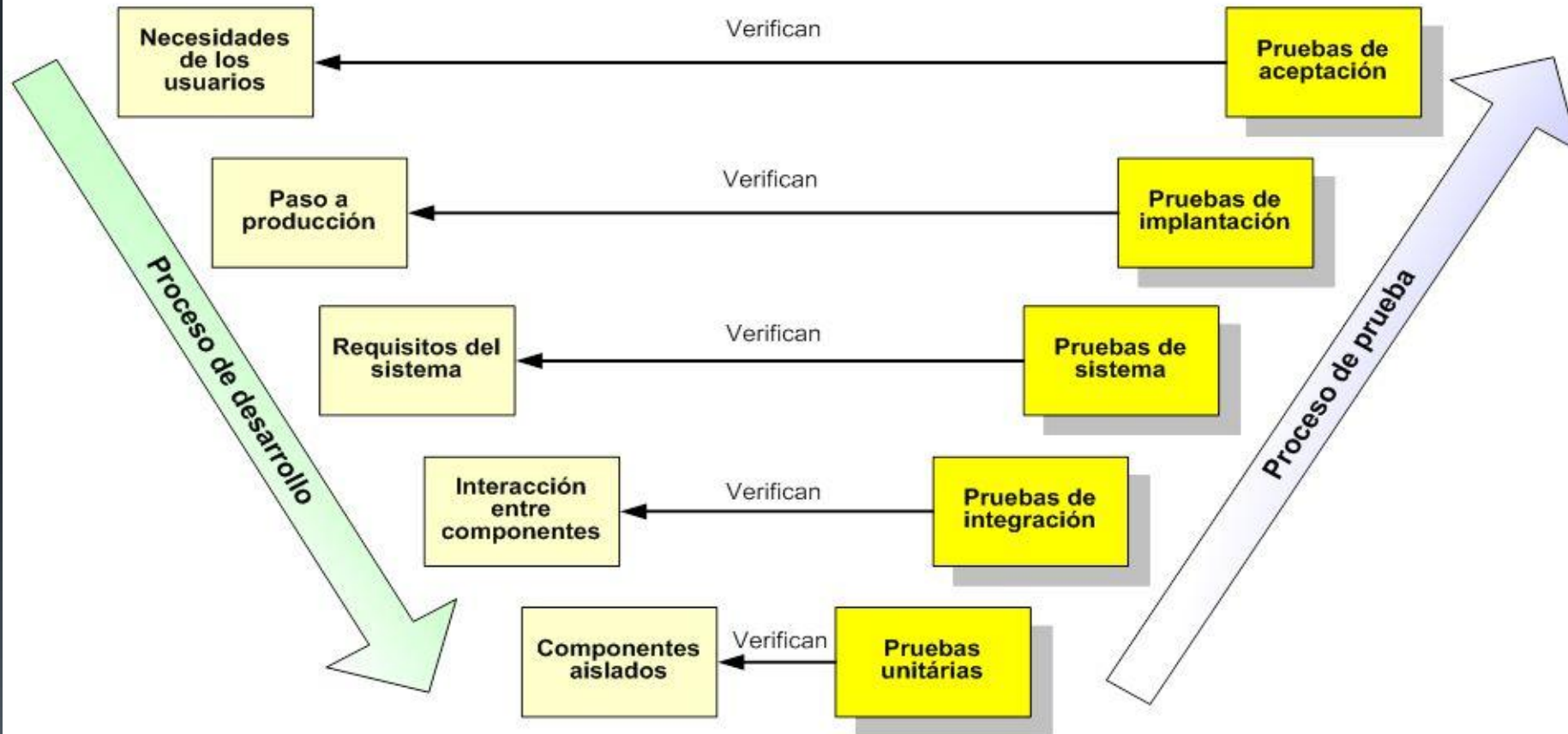
- Especificación del diseño de prueba
- Especificación de los casos de prueba
- Especificación del procedimiento
- Reporte de avance de los ciclos probados
- Registro de la prueba
- Reporte de incidentes Sumario de la prueba

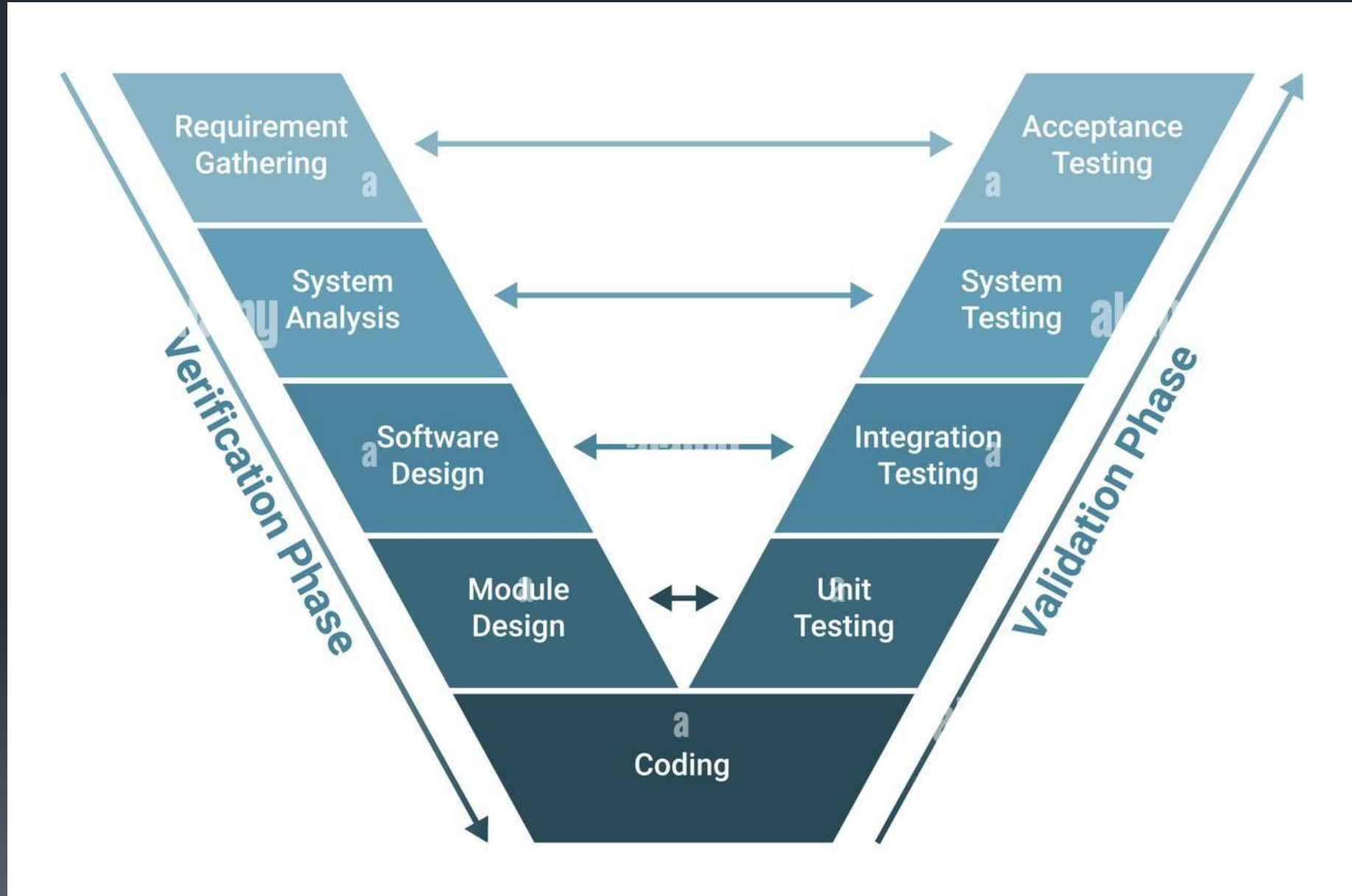
# ***ESTÁNDARES INTERNACIONALES***

- TPI (Test Process Improvement)
  - Metodología para evaluar el nivel de madurez de pruebas en una organización, así como para mejorar el proceso.



# Niveles de prueba





# *Tarea para el hogar!*

Buscar un defecto en algún software que utilices a menudo, escríbelo y prepárate para contarnos en clase!

¿Qué piensa que falta probar para salvar ese defecto?

¿Ese defecto genera un fallo en el software?

¿Crees que el error fue de diseño o de desarrollo?



***FIN***

Materia: Metodología de Pruebas  
Profesor: Ing. Pablo Andrés Pérez