¿Qué es “Testing”?

En el mundo del desarrollo de software

Es un conjunto de actividades que buscan **verificar** y **validar** que una aplicación **cumple** con sus **requisitos** y **funciona** correctamente, ayudando a **prevenir** defectos y minimizar riesgos.

OBJETIVOS

Evaluar requisitos, historias de usuario, diseños y código.

Provocar fallos y encontrar defectos.

Asegurar la cobertura necesaria del objeto de prueba.

Reducir riesgos y costos asociados a fallos en producción.

Verificar si se han cumplido los requisitos especificados y que cumpla con requisitos contractuales, legales y regulatorios.

Proporcionar información a los interesados para que puedan tomar decisiones informadas.

Generar confianza en la calidad del objeto de prueba.

Validar si el objeto de prueba está completo y funciona como lo esperan los interesados.

Creado por Fransa J. Aravena

Fuente: Syllabus ISTQB v4.0.1

Conceptos clave

Error

Acción humana que produce un resultado incorrecto, por malentendidos, falta de atención o información.
Ej: Un desarrollador malinterpreta un requisito y escribe en el código una condición equivocada.

Defecto (Bug)

Imperfección en un sistema que puede causar fallos.
Ej: El error en el código genera un comportamiento inesperado en una funcionalidad.

Fallo (Failure)

Evento en el cuál un sistema no realiza una función requerida.
Ej: Al ejecutar el programa con ciertos datos, el sistema deja de funcionar.

Caso de Prueba

Conjunto de condiciones diseñadas para verificar el sistema.

Plan de pruebas

Documento que describe el alcance, enfoque, recursos y cronograma de las actividades de prueba previstas.

Cobertura de pruebas

Porcentaje del producto que ha sido probado.
NO garantiza calidad del software, ya que indica QUÉ se probó pero no QUÉ TAN BIEN se probó.

Creado por Fransa J. Aravena

Fuente: Syllabus ISTQB v4.0.1

Técnicas de Prueba

Categorías principales:

Pruebas de Caja Negra

Basadas en la especificación, sin conocer el código.
Se enfocan en entradas y salidas.
Ej: Probar el login con credenciales válidas e inválidas.

Pruebas de Caja Blanca

Basadas en la estructura interna, la lógica y el flujo del código.
Ej: *Probar todas las ramas de un condicional (if/else).*

Prueba Basadas en la Experiencia

Dependen del conocimiento y experiencia del tester para descubrir defectos.
Ej: Explorar una app sin casos definidos, buscando fallos.

Según el Objetivo

Pruebas Funcionales

Evalúan si un componente o sistema satisface los requisitos funcionales.
Ej: Verificar que el total del carrito se actualice al agregar productos.

Pruebas No Funcionales

Evalúan cómo funciona el sistema: rendimiento, seguridad, usabilidad, etc.
Ej: Medir el tiempo de respuesta con 1.000 usuarios concurrentes.

Creado por Fransa J. Aravena

Fuente: ISQTB Glossary

Tipos de prueba

Pruebas de Humo

Conjunto inicial de pruebas para verificar que las funciones críticas del sistema funcionan correctamente.

Pruebas de Sanidad

Verificación rápida y focalizada asegurando que una función específica o un defecto corregido trabaja como se espera.

Pruebas de Regresión

Confirman que las modificaciones no hayan introducido nuevos defectos en otras áreas del software.

Pruebas Exploratorias

El tester diseña, ejecuta y aprende sobre el sistema simultáneamente, adaptando las pruebas según lo descubierto.

Pruebas de Rendimiento

Evalúan el tiempo de respuesta y estabilidad del sistema bajo una carga definida.

Pruebas de Estrés

Analizan el comportamiento del sistema más allá de sus límites normales, hasta el punto de falla.

cypress

Se

Playwright

Katalon

TestComplete

Ghost Inspector

POSTMAN

APACHE JMeter™

k6

Gatling

LOCUST

Load Runner

perforce BlazeMeter

Creado por Fransa J. Aravena

Fuente: Syllabus ISTQB v4.0.1

Pruebas clasificadas por nivel (según SDLC)

Pruebas Unitarias

Prueban componentes **individuales** en aislamiento, normalmente realizadas por los desarrolladores.

Ej: *Verificar que una función calcule correctamente el total de un pedido.*

Pruebas Integración

Evalúan las interfaces y la **interacción** entre módulos o sistemas.

Ej: *Comprobar que el módulo de pagos se conecte correctamente con PayPal.*

Pruebas de Sistema

Validan el sistema completo e integrado frente a los requisitos especificados.

Ej: *Probar un e-commerce: búsqueda, carrito, pago y confirmación.*

Pruebas de Aceptación

Confirman que el sistema cumpla los criterios del negocio y necesidades del usuario.

Ej: *Validar que un sistema de reservas funcione según lo acordado con el cliente.*

Creado por Fransa J. Aravena

Fuente: Syllabus ISTQB v4.0.1

LOS 7 PRINCIPIOS DEL TESTING

Las pruebas muestran la presencia de defectos, no la ausencia: Que no se encuentren defectos no significa que el software esté libre de errores.

Las pruebas exhaustivas son imposibles: No se puede probar todo. Se usan técnicas, priorización y pruebas basadas en riesgos.

Detectar a tiempo es más barato: Detectar defectos en etapas iniciales reduce costos, tiempo y evita fallos en producción.

Los defectos tienden a agruparse: Unos pocos módulos suelen contener la mayoría de los defectos.

Las pruebas se desgastan: Repetir siempre los mismos tests los hace menos efectivos para encontrar nuevos fallos.

Las pruebas dependen del contexto: No existe un enfoque único. Las pruebas deben adaptarse al proyecto y sus objetivos.

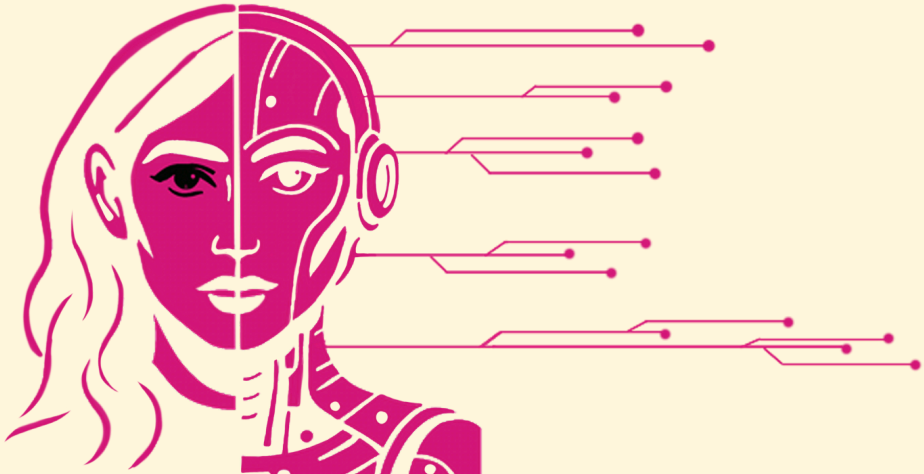
Falacia de ausencia de defectos: Un software sin defectos no siempre es exitoso si no satisface las necesidades del usuario.

Creado por Fransa J. Aravena

Fuente: Syllabus ISTQB v4.0.1

Conectando y apoyando a mujeres en Latam para crecer juntas en el camino del Testing

Mujeres Testing - Latam



Creado por Fransa J. Aravena

2025