

A photograph of a female scientist with brown hair tied back, wearing clear safety goggles with orange frames and a white lab coat. She is looking down at a white notebook, holding a blue pen and writing. In the background, there is a rack of test tubes containing various colored liquids (yellow, green, blue) and a white microscope on a stand. The lighting is bright and focused on her face and hands.

# La importancia del testing de software

El testing de software es esencial para detectar errores, defectos o fallas en un sistema o aplicación informática. Garantiza la calidad del software y reduce los riesgos asociados con su uso, lo que ahorra tiempo, dinero y evita problemas para los usuarios finales.



**by Matias Ruiz**

# Diferencias entre pruebas funcionales y no funcionales

## 1 — Pruebas Funcionales

Se centran en verificar si el software cumple con los requisitos funcionales, es decir, si realiza las acciones esperadas y produce los resultados correctos.

## 2 — Pruebas No Funcionales

Se centran en la seguridad, rendimiento, escalabilidad y usabilidad del software, aspectos no directamente relacionados con su funcionalidad.

# Ciclo de vida del testing de software

## Planificación

Establecer los objetivos, identificar los recursos necesarios y definir una estrategia de pruebas clara.

## Implementación

Realizar pruebas manuales y/o automatizadas para detectar errores, defectos o fallas en el sistema.

## Corrección de Defectos

Reportar y solucionar los defectos encontrados durante el proceso de testing para garantizar la calidad del software.

## Cierre

Generar informes periódicos que resuman el estado de los defectos encontrados durante las pruebas.

# Diferentes tipos de pruebas de software

## Pruebas de Integración

Se realizan para verificar el correcto funcionamiento de las diferentes partes integradas del sistema.

## Pruebas de Rendimiento

Se realizan para verificar que el software cumple con los requisitos de rendimiento esperados.

## Pruebas de Usabilidad

Se realizan para verificar que el software es fácil de usar y aprender por los usuarios finales.



## Pruebas Unitarias

Se realizan en las partes más pequeñas e independientes del sistema, verificando su correcto funcionamiento.

## Pruebas de Aceptación

Se realizan para validar que el software cumple con los requisitos del cliente o usuario final.

## Pruebas de Seguridad

Se realizan para asegurar que el software es seguro y protege adecuadamente los datos.

# Diferencias entre las pruebas de caja negra y caja blanca



Pruebas de Caja Negra

Se evalúa el software sin conocer la estructura interna o detalles de implementación.



Pruebas de Caja Blanca

Se basa en el conocimiento de la estructura interna del software y en su lógica de funcionamiento.

# Errores, defectos y fallos en el testing de software

1 Error

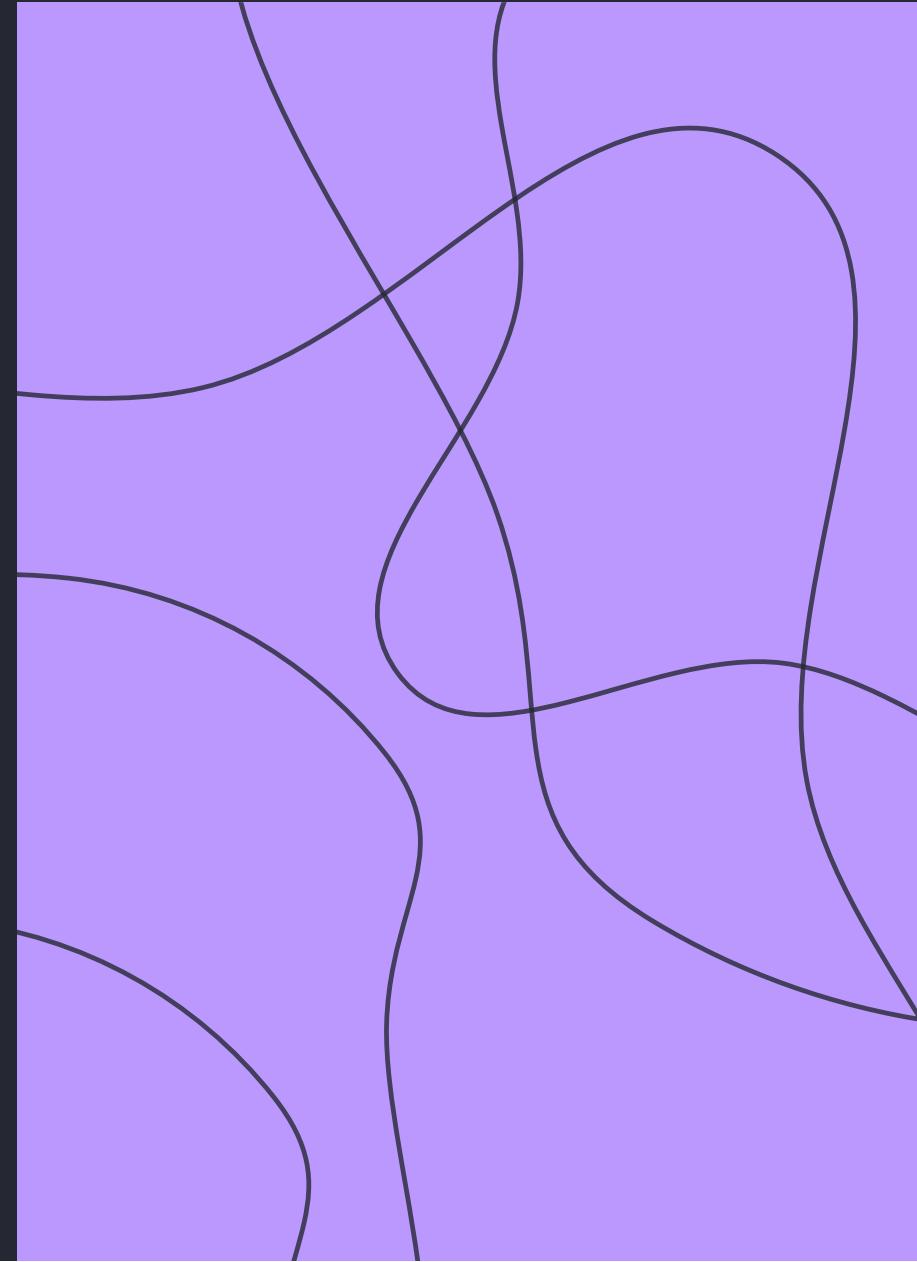
Es una acción humana que produce un resultado incorrecto o inesperado.

2 Defecto

Es una anomalía o desviación del software con respecto a sus requisitos o especificaciones.

3 Fallo

Ocurre cuando el software no realiza las funciones esperadas o produce resultados incorrectos durante su ejecución.



# Cobertura de pruebas y criterios comunes

## Cobertura de Instrucciones

Evalúa la cantidad de instrucciones ejecutadas por las pruebas.

## Cobertura de Ramificacióne

S  
Evalúa la cantidad de posibles ramificaciones en el código que las pruebas han cubierto.

## Cobertura de Condiciones

Evalúa la cantidad de condiciones lógicas del programa que las pruebas han cubierto.

## Cobertura de Bucles

Evalúa la cantidad de iteraciones en bucles que las pruebas han cubierto.

# Automatización de pruebas y sus beneficios

1

## Ahorro de Tiempo y Recursos

Al automatizar las pruebas, se reducen significativamente los costos y el tiempo necesarios para verificar el software.

2

## Mayor Cobertura

La automatización permite realizar pruebas en diferentes plataformas y entornos, lo que significa que se pueden realizar más pruebas en menos tiempo.

3

## Repetibilidad

Las pruebas automatizadas se pueden repetir constantemente sin errores humanos, lo que garantiza la integridad y validez de los resultados.

4

## Precisión

A diferencia de las pruebas manuales, las pruebas automatizadas son precisas y consistentes en la repetición de pruebas.

# Buenas prácticas en el proceso de testing de software



## Comenzar el testing temprano

Desde el inicio del proceso de desarrollo, el equipo de testing puede colaborar con el equipo de desarrollo para garantizar que los requisitos se cumplan de manera adecuada.



## Definir una estrategia de pruebas clara

Establecer objetivos claros, tipos de pruebas y estrategias para cada aplicación.

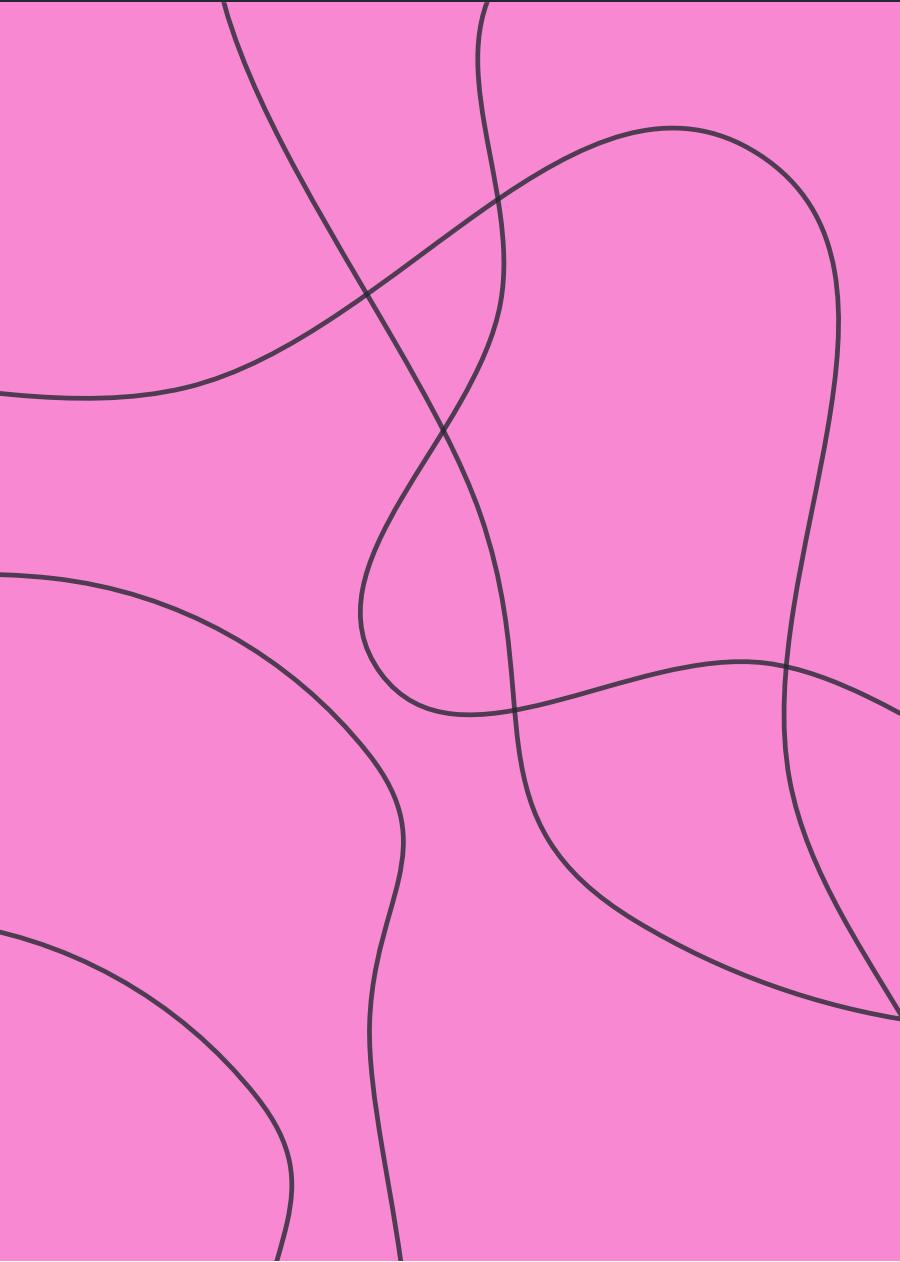


## Documentación adecuada

Llevar registros precisos de la ejecución y resultados de las pruebas para la validación y verificación.

## Diseñar casos de prueba efectivos

Las pruebas deben ser relevantes, económicas, comprensibles y repetibles.



## Registro de defectos

Registra cada defecto encontrado en un sistema de seguimiento de incidentes o herramienta similar.

## Priorización y clasificación

Evalúa la gravedad y la importancia de cada defecto y asígnale una prioridad y una severidad



# Comunicación con el equipo de desarrollo

Comunica los defectos al equipo de desarrollo de manera clara y precisa

## Seguimiento y resolución

Realiza un seguimiento regular de los defectos reportados para asegurarte de que se están abordando y corrigiendo

## Verificación y cierre

Una vez que un defecto ha sido corregido, verifica su resolución y realiza pruebas adicionales para asegurarte de que no haya introducido nuevos problemas

## Generación de informes

Genera informes periódicos que resuman el estado de los defectos encontrados durante las pruebas

