

Pruebas estáticas y dinámicas

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. [Pruebas estáticas](#)
2. [Pruebas dinámicas](#)
3. [Conclusión](#)

“

Las pruebas estáticas y dinámicas tienen el objetivo de proporcionar una **evaluación de calidad** de los productos de trabajo e **identificar defectos** en forma temprana.



”

1 | Pruebas estáticas

Conceptos básicos de la prueba estática

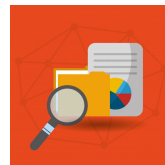
La prueba estática se basa en la **evaluación manual** de los productos de trabajo (es decir, revisiones) o en la evaluación basada en herramientas del código u otros productos de trabajo (es decir, análisis estático). Este tipo de pruebas no requieren la ejecución del software que se está probando.



Conceptos básicos de la prueba estática

Se utilizan este tipo de pruebas para examinar cualquier producto de trabajo, como por ejemplo:

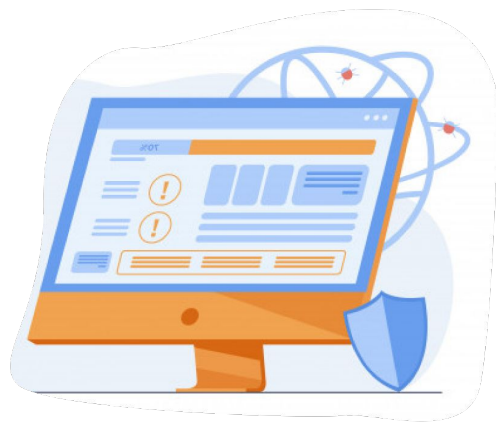
- Especificaciones, requisitos de negocio, funcionales y de seguridad.
- Épicas, historias de usuarios y criterios de aceptación.
- Especificaciones de arquitectura y diseño.
- Código.
- Productos de prueba: planes, casos, procedimientos y guiones de prueba.
- Manuales de usuario.
- Contratos, planes de proyecto, calendarios y presupuestos.



Ventajas de las pruebas estáticas tempranas

Cuando se aplica al principio del ciclo de vida del desarrollo del software, la prueba estática permite la detección temprana de defectos. Esto genera una reducción de costos y tiempo de desarrollo y prueba.

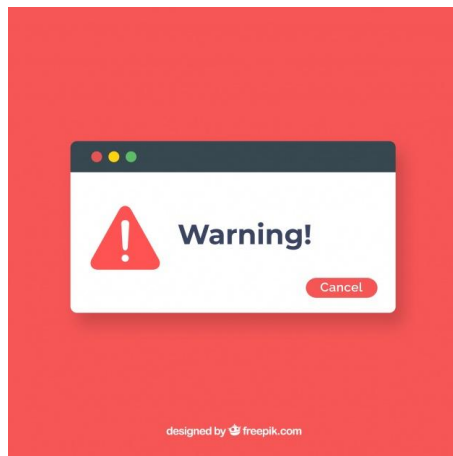
Por el contrario, si el defecto se encuentra luego de las pruebas dinámicas, solucionarlo va a requerir el cambio de código, realizar una prueba de confirmación y luego incluir el mismo en pruebas de regresión, además de los cambios de toda la documentación asociada.



Defectos encontrados con pruebas estáticas

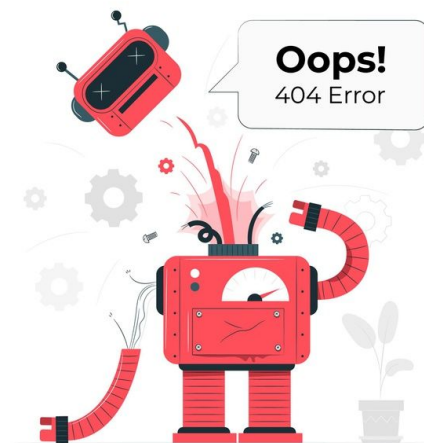
Algunos de los defectos encontrados con pruebas estáticas que son más fáciles y económicos de detectar y corregir son:

- Defectos en los requisitos (inconsistencias, ambigüedades, etc.).
- Defectos de diseño (estructura de base de datos ineficiente, alto acoplamiento, etc.).
- Defectos de codificación (variables con valores no definidos, código inalcanzable o duplicado, etc.).
- Desviaciones con respecto a estándares (falta de uso de estándares de codificación).



Defectos encontrados con pruebas estáticas

- Especificaciones de interfaz incorrectas (unidades de medida diferente, etc.).
- Vulnerabilidades de seguridad (susceptibilidad a desbordamiento de la memoria intermedia).
- Diferencias o inexactitudes en la trazabilidad o cobertura de la base de prueba (falta de pruebas para un criterio de aceptación).
- Defectos de mantenibilidad (mala reutilización de componentes, modularización inadecuada, etc.).



2 | Pruebas Dinámicas

Conceptos básicos de la prueba dinámica

Las pruebas dinámicas requieren la ejecución del software, componente o sistema.

Se complementan con las pruebas estáticas debido a que encuentra diferentes tipos de defectos. Para la generación de casos de prueba se utilizan diferentes técnicas de caja negra, caja blanca o basadas en la experiencia de usuario.



Conceptos básicos de la prueba dinámica



Durante las pruebas dinámicas se ejecuta el software utilizando un conjunto de valores de entrada y su resultado se analiza y compara con el resultado esperado.

Las fallas más comunes encontradas con este tipo de pruebas son:

- Fallas de funcionalidad.
- Fallas de interacción entre módulos.
- Fallas de rendimiento y seguridad.

3 | Conclusión

Conclusión

Prueba estática	Prueba dinámica
Se basa en la evaluación manual mediante revisiones y análisis estático	Requieren la ejecución del software, componente o sistema
Detecta los defectos en productos de trabajo.	Detecta los defectos y fallas cuando se ejecuta el software.
Se centra en mejorar la consistencia y la calidad de los productos de trabajo.	Se centra en los comportamientos visibles desde el exterior.
El costo de solucionar un defecto es menor	El costo de solucionar un defecto es mayor

DigitalHouse>
Coding School