

# Técnicas estáticas y dinámicas

**Definición:** Las técnicas estáticas de prueba se centran en el análisis y revisión de documentos y código fuente sin ejecutar el software. Estas técnicas se aplican durante las primeras etapas del ciclo de vida del desarrollo de software y son fundamentales para detectar problemas y mejorar la calidad del producto.

Por otro lado, las técnicas dinámicas de prueba implican la ejecución del software y la observación de su comportamiento en tiempo de ejecución. Estas técnicas se utilizan para validar el cumplimiento de los requisitos funcionales y no funcionales, y para identificar errores y comportamientos inesperados.

Diferencias entre técnicas estáticas y dinámicas:

A continuación, se muestra una tabla comparativa que resalta las diferencias clave entre las técnicas estáticas y dinámicas de prueba:

Técnicas Estáticas	Técnicas Dinámicas
Análisis y revisión sin ejecución	Ejecución del software
Se aplica antes de la ejecución	Se aplica durante la ejecución
Detecta problemas en documentos y código fuente	Valida requisitos funcionales y no funcionales
Ejemplos: revisión de documentos, inspecciones de código	Ejemplos: pruebas de unidad, pruebas de integración

## Aplicación de técnicas estáticas para la revisión y análisis de

**documentación y código fuente:** Las técnicas estáticas son ampliamente utilizadas para revisar y analizar la documentación y el código fuente. Algunas de las técnicas estáticas comunes incluyen:

1. **Revisión de documentos:** Consiste en examinar documentos, como especificaciones de requisitos, diseños y casos de uso, para identificar problemas y discrepancias. Ejemplo: revisar una especificación de requisitos para asegurarse de que todos los requisitos estén correctamente definidos y sean consistentes.
2. **Inspección de código:** Implica un examen minucioso del código fuente para identificar errores, violaciones de estándares de codificación y posibles problemas de rendimiento. Ejemplo: revisar el código fuente de un módulo para identificar variables no utilizadas o secciones de código redundantes.

## **Aplicación de técnicas dinámicas para la ejecución de pruebas en tiempo de ejecución:**

Las técnicas dinámicas se utilizan para ejecutar pruebas en tiempo de ejecución y validar el comportamiento del software. Algunas de las técnicas dinámicas comunes son:

1. **Pruebas de unidad:** Se enfocan en probar componentes individuales del software, como funciones o métodos, para garantizar que funcionen correctamente. Ejemplo: diseñar casos de prueba para una función específica que valida el ingreso de datos en un formulario.
2. **Pruebas de integración:** Se centran en probar la interacción y la integración entre diferentes componentes del software para garantizar que funcionen juntos sin problemas. Ejemplo: ejecutar pruebas para verificar que los módulos de un sistema se comuniquen correctamente entre sí.

Es importante destacar que existen numerosas técnicas estáticas y dinámicas disponibles, y que la elección de las técnicas adecuadas dependerá del contexto y los objetivos específicos del proyecto de desarrollo de software. Aquí tienes algunos ejemplos adicionales de técnicas estáticas y dinámicas:

### **Ejemplos de técnicas estáticas:**

1. **Análisis de trazabilidad:** Consiste en verificar la trazabilidad de los requisitos a lo largo de todo el ciclo de vida del desarrollo de software. Se busca asegurar que todos los requisitos estén cubiertos y que no haya requisitos ambiguos o redundantes.
2. **Análisis de impacto:** Se utiliza para evaluar el impacto de posibles cambios en el software. Permite identificar las áreas que se verán afectadas por los cambios propuestos y evaluar los riesgos asociados.

### **Ejemplos de técnicas dinámicas:**

1. **Pruebas de aceptación:** Son pruebas realizadas por los usuarios finales o el cliente para validar que el software cumple con sus requisitos y expectativas. Estas pruebas se centran en escenarios del mundo real y se ejecutan en un entorno similar al de producción.

2. **Pruebas de carga:** Se realizan para evaluar el rendimiento del software bajo condiciones de carga intensiva. Estas pruebas se enfocan en medir la capacidad del sistema para manejar un alto volumen de usuarios o transacciones y detectar posibles cuellos de botella.

### Formatos y documentos utilizados en las técnicas estáticas y dinámicas:

En las técnicas estáticas y dinámicas, se utilizan varios formatos y documentos para registrar y comunicar los resultados de las pruebas. Algunos de los formatos comunes incluyen:

1. **Informes de revisión:** Documento que resume los hallazgos y recomendaciones resultantes de una revisión de documentos o código fuente.

Fecha	Responsable	Documento Revisado	Hallazgos	Recomendaciones
10/05/2023	Juan Pérez	Especificación de Requisitos	Requisito 2.3 mal redactado	Revisar y corregir el requisito 2.3 para una mayor claridad y coherencia.
10/05/2023	Ana Gómez	Código Fuente	Variables no utilizadas en el archivo "main.py"	Eliminar las variables no utilizadas para mantener un código limpio y eficiente.

En este ejemplo, se registra la fecha de la revisión, el responsable de la revisión, el documento o código fuente revisado, los hallazgos encontrados durante la revisión y las recomendaciones correspondientes.

2. **Casos de prueba:** Documentos que describen los escenarios de prueba, los pasos a seguir, los datos de entrada y los resultados esperados. Estos casos de prueba se utilizan para guiar la ejecución de las pruebas y registrar los resultados obtenidos.

Caso de Prueba	Descripción	Pasos a Seguir	Datos de Entrada	Resultado Esperado
CP-001	Prueba de inicio de sesión	1. Abrir la página de inicio de sesión.	Usuario: "usuario1"	Se inicia sesión correctamente en la cuenta del usuario.
CP-002	Prueba de registro de usuario	1. Abrir la página de registro.	Nombre: "John Doe", Email: " <a href="mailto:john@example.com">john@example.com</a> "	Se registra exitosamente el usuario en la base de datos.
CP-003	Prueba de búsqueda de producto	1. Ingresar un término de búsqueda en el campo de búsqueda.	Término de Búsqueda: "smartphone"	Se muestran los resultados de búsqueda relacionados.

En este ejemplo, se presenta una tabla con diferentes casos de prueba. Cada caso de prueba incluye una descripción, los pasos a seguir durante la ejecución de la prueba, los datos de entrada necesarios y el resultado esperado después de ejecutar la prueba.

3. **Matriz de trazabilidad:** Tabla que muestra la relación entre los requisitos, los casos de prueba y los resultados de las pruebas. Ayuda a garantizar que todos los requisitos estén cubiertos por los casos de prueba y que se registren los resultados de las pruebas correspondientes.

Requisito	Caso de Prueba	Resultado de Prueba
RQ-001	CP-001	Pasó
RQ-002	CP-002	Falló
RQ-003	CP-003	Pasó
RQ-004	CP-004	Pendiente

En esta tabla, se muestra la relación entre los requisitos (identificados por RQ-001, RQ-002, etc.), los casos de prueba correspondientes (identificados por CP-001, CP-002, etc.) y los resultados de las pruebas. Los resultados de las pruebas pueden indicar si el caso de prueba pasó, falló o está pendiente.

Esta matriz de trazabilidad permite tener una visión clara y estructurada de cómo se cubren los requisitos mediante los casos de prueba y cómo se registran los resultados de las pruebas realizadas. De esta manera, se garantiza que todos los requisitos estén cubiertos y se tenga un seguimiento adecuado de los resultados obtenidos.

4. **Informes de resultados de prueba:** Documentos que resumen los resultados de las pruebas ejecutadas, incluyendo las fallas encontradas, los problemas identificados y las mejoras propuestas.

Número de Prueba	Descripción de la Prueba	Estado	Fallas Encontradas	Problemas Identificados	Mejoras Propuestas
1	Prueba de inicio de sesión	Exitosa	-	-	-
2	Prueba de registro de usuario	Fallida	Error al validar el formato del campo de correo electrónico	-	Actualizar la validación del formato del campo de correo electrónico
3	Prueba de búsqueda de producto	Exitosa	-	Lentitud en la carga de resultados de búsqueda	Optimizar la velocidad de carga de los resultados de búsqueda

En este ejemplo, se presenta una tabla que resume los resultados de las pruebas ejecutadas. Cada prueba se identifica con un número y se proporciona una descripción de la prueba. El estado indica si la prueba fue exitosa o fallida. En caso de haber fallas encontradas, se detallan en la columna correspondiente. Además, se identifican los problemas que se han identificado durante las pruebas y se proponen mejoras para abordar esos problemas.