Clase 03. ¿Cuáles son los distintos tipos de testing?

# Introducción a los tipos de pruebas y niveles de prueba

## Introducción

Todas las pruebas dinámicas poseen un tipo y un nivel. Ambas características pueden cruzarse, para que cada tipo de prueba pueda darse en todos los niveles de prueba.

## Tipos de testing

### Tipos de prueba

Es un grupo de actividades de prueba destinadas a probar características específicas de un sistema de software, o de una parte de un sistema, basadas en objetivos de prueba específicos.

Tipos de pruebas:

* Funcional
* No funcional
* Caja blanca
* Asociada al cambio

### 4 Objetivos de los tipos de pruebas

* Evaluar la completitud, corrección y pertinencia - **Funcional.**
* Evaluar la viabilidad, eficiencia de desempeño, seguridad, compatibilidad y usabilidad - **No Funcional.**
* Evaluar si la estructura del sistema es correcta, completa y según lo especificado - **Caja Blanca.**
* Evaluar los efectos de los cambios, tales como confirmar que los defectos han sido corregidos y buscar cambios no deseados en el comportamiento que resultan de cambios en el software o en el entorno - **Asociada al Cambio.**

### Prueba funcional

#### Definición

La prueba funcional de un sistema incluye pruebas que evalúan las funciones que el sistema debe realizar. Los requisitos funcionales pueden estar descritos en productos de trabajo tales como especificaciones de requisitos de negocio, épicas, historias de usuario, casos de uso, o especificaciones funcionales.

Las funciones describen "qué" debe hacer el sistema.

#### ¿Cuándo se hacen las pruebas funcionales?

Se deben realizar pruebas funcionales en todos los niveles de prueba, aunque el enfoque es diferente en cada nivel.

#### Intensidad pruebas funcionales

Se puede medir la intensidad de la prueba funcional a través de la cobertura funcional. La cobertura funcional es la medida en que algún tipo de elemento funcional ha sido practicado por pruebas, y se expresa como un porcentaje del tipo o tipos de elementos cubiertos.

#### Alcance pruebas funcionales

El diseño y la ejecución de pruebas funcionales pueden implicar competencias o conocimientos especiales, como el conocimiento del problema de negocio específico que resuelve el software o el papel particular que desempeña el software.

### Prueba No Funcional

#### Definición

Las pruebas no-funcionales evalúan otras características de sistemas, como por ejemplo el UX/UI, conectividad o la seguridad. La prueba no-funcional prueba que tan bien se comporta un sistema básicamente. Hacen foco en los Requerimientos No Funcionales.

#### Requerimientos no funcionales

Son los requerimientos que especifican propiedades del sistema tales como:

* Rendimiento
* Seguridad
* Disponibilidad
* Performance
* Accesibilidad
* Estabilidad, etc.

#### Ejemplos

* El sistema estará restringido bajo contraseñas cifradas y usuarios definidos. -Seguridad
* El sistema deberá tener interfaces intuitivas y deberan respetar los colores azul y naranja.-Usabilidad
* El sistema no debe tardar más de 10 segundos en mostrar resultados de una búsqueda.-Performance

#### ¿Cuándo se hacen pruebas no funcionales?

Se sugiere que las pruebas no-funcionales se realicen en cada nivel de prueba ya que pueden prever problemas en los sistemas que van por fuera de los errores encontrados en pruebas funcionales. Encontrar problemas del tipo no-funcional sobre el final de un desarrollo, puede jugar un papel crucial en la implementación del mismo.

#### Intensidad pruebas no funcionales

Tal como en las pruebas funcionales, acá también se puede medir la intensidad de una prueba no funcional a través de la cobertura no funcional. También se mide de acuerdo a la medida de cómo algún objeto no funcional ha sido probado.

Por ejemplo, se puede hacer una trazabilidad entre las pruebas y los sistemas operativos compatibles con una aplicación, se puede calcular la cantidad de sistemas operativos trabajados en las pruebas de compatibilidad, identificando el porcentaje de carencia de pruebas.

#### Alcance pruebas no funcionales

Para este tipo de pruebas, se debe contar con ciertos conocimientos o competencias especiales al momento de hacer el diseño y la ejecución de las mismas.

Fuente: Foto de Pexels

\*Nota: Dentro de un par de clases, vamos a ampliar estos conceptos

Ejemplo: buscamos hacer click en un botón y no funciona. Creemos que es un problema en el código, pero en realidad es un error de conexión.

### Prueba de caja blanca

#### Definición

Se obtienen pruebas leyendo la infraestructura del sistema, el código del mismo, en su implementación o arquitectura. También incluye flujos de datos, bases de datos o flujos de trabajo. Esta es la razón por la que es un tipo de prueba que necesita un seniority más avanzado.

#### Intensidad pruebas de caja blanca

Este tipo de pruebas también lleva una medición de intensidad a través de la cobertura estructural. En esta, la medida se toma a partir de los objetos estructurales que se hayan realizado contra lo que resta probar. Dependiendo qué objeto sea el que se esté probando es el tipo de porcentaje que se aplicará.

#### Conocimiento de código

Claramente las pruebas del tipo de caja blanca necesitan de un tipo de conocimiento relacionado a otras áreas que van por fuera del tester QA. Su diseño y ejecución va a estar basado de las competencias que se tenga sobre el tema.

Más allá de los conocimientos, se recomienda contar con el soporte de un experto del área durante estas pruebas.

### Prueba asociada al cambio

#### Definición

Como su nombre lo indica, son pruebas que se utilizan para comprobar corrección de bugs o mejoras en un sistema o nuevas funcionalidades, para poder asegurar que el sistema funciona como corresponde.

Aquí es donde entra en juego el papel del QA de Automatización.

#### Áreas de las pruebas

Las vamos a separar en dos grandes áreas:

* Pruebas de confirmación.
* Pruebas de regresión.

##### Pruebas de confirmación

Cuando nos entregan el sistema corregido de algún defecto, se llevan adelante estas pruebas para asegurar que el error ya no se encuentra. Esto implica realizar los pasos que llevaron adelante ese error, y continuar con los próximos pasos, si es que este error bloqueo alguno o probar si otras áreas del sistema, relacionadas con el bug, siguen funcionando de forma correcta.

##### Pruebas de regresión

Como su nombre lo indica, se realizará una regresión sobre lo que ya se probó. Esta prueba se utiliza para corroborar que ninguna otra área del sistema haya sido afectada por el cambio realizado en él. Esto es para ver que no haya ningún defecto como daño secundario a causa de estos cambios.

#### ¿Cuándo se aplican estas pruebas?

Estas dos pruebas se realizan en todos los niveles de prueba, especialmente en metodologías del tipo Agile, debido a la gran cantidad de iteraciones incrementales en sus ciclos de desarrollo.

#### Aplicación de estas pruebas

Como se deben imaginar, acá es donde la labor de los QA de automatización, o como algunos los llaman, los QAA, tienen un papel importantísimo. Sin las pruebas de regresión automatizadas, el tiempo de estas pruebas pasaría por encima de los tiempos de las demás. Generando que la dependencia del QA sea entera en volver a probar funcionalidades para ver que no pierdan compatibilidad.

## Niveles de testing

### Introducción a niveles de testing

Los niveles de prueba son grupos de actividades de prueba que se organizan y gestionan conjuntamente.

Cada nivel de prueba es una instancia del proceso de prueba realizadas en relación con el software en un nivel de desarrollo determinado, desde unidades o componentes individuales hasta sistemas completos.

### 4 Niveles de testing

* Prueba de aceptación
* Prueba de sistema
* Prueba de integración
* Prueba de componente

### Características de los niveles de testing

* Objetivos específicos.
* Bases de prueba, referenciadas para generar casos de prueba.
* Objeto de prueba (es decir, lo que se está probando).
* Defectos y fallos característicos.
* Enfoques y responsabilidades específicos.

### Prueba de componente

#### Definición

La prueba de componente (también conocida como prueba unitaria o de módulo) se centra en los componentes que se pueden probar por separado.

#### Objetivo prueba de componente

* Reducir el riesgo.
* Verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados.
* Generar confianza en la calidad del componente.
* Encontrar defectos en el componente.
* Prevenir la propagación de defectos a niveles de prueba superiores.

#### ¿Cómo se aplica?

La prueba de componente, a menudo, se realiza de forma aislada del resto del sistema, dependiendo del modelo de ciclo de vida de desarrollo de software y del sistema, lo que puede requerir objetos simulados, virtualización de servicios, arneses, stubs y controladores.

#### Las pruebas de componente pueden cubrir:

1. Funcionalidad. Por ejemplo, la exactitud de los cálculos
2. Las características no funcionales. Por ejemplo, la búsqueda de fugas de memoria
3. Las propiedades estructurales. Por ejemplo, pruebas de decisión.

#### Aplicación

Se suele aplicar como base de pruebas para componentes como diseño detallado, código, modelo de datos o especificaciones de componentes.

Por esto mismo, los objetos de prueba suelen ser directamente código como componentes, unidades, módulos, clases, estructura de datos, módulos de bases de datos, etc...

#### Enfoque

En general, el desarrollador hace las pruebas de componentes utilizando directamente su código base. También escriben y ejecutan pruebas después de haber escrito el código de un componente.

De esta forma, se suele utilizar el mismo código para recrear las pruebas de una forma automatizada.

#### Prueba de componente en TDD

Considerar el desarrollo guiado por pruebas (Test Driven Development, TDD por sus siglas en inglés).

El TDD es altamente iterativo y se basa en ciclos de desarrollo de casos de prueba

automatizados, luego se construyen e integran pequeños fragmentos de código, a continuación, se ejecuta la prueba de componente, se corrige cualquier cuestión y se refactoriza el código.

### Prueba de integración

#### Definición

La prueba de integración se centra en las interacciones entre componentes o sistemas.

Cuando se integran las partes que se probaron aisladamente, se prueba todo en conjunto validando su funcionamiento como un todo.

#### Objetivos de prueba de integración

* Reducir el riesgo.
* Verificar que los comportamientos funcionales y no funcionales de las interfaces sean los diseñados y especificados.
* Generar confianza en la calidad de las interfaces.
* Encontrar defectos (que pueden estar en las propias interfaces o dentro de los componentes o sistemas).
* Prevenir la propagación de defectos a niveles de prueba superiores.

Nota: Son los mismos objetivos que las pruebas de componente, pero a un nivel más alto.

#### Aplicación

Se suele aplicar como base de pruebas para componentes como diseño de software y sistemas, casos de uso, flujos de trabajo o diagramas de secuencia.

Por esto mismo, los objetos de prueba suelen ser subsistemas, bases de datos, infraestructura, interfaces, microservicios, etc...

#### Enfoque

La prueba debe centrarse en la comunicación entre los módulos, no en la funcionalidad de los módulos individuales, cómo debería haberse hecho durante la prueba de componente.

La prueba de integración de componentes suele ser responsabilidad de los desarrolladores.

La prueba de integración de sistemas es, en general, responsabilidad de los testers.

### Prueba de sistema

#### Definición

La prueba de sistema se centra en el comportamiento y las capacidades de todo un sistema o producto, a menudo teniendo en cuenta las tareas extremo a extremo que el sistema puede realizar y los comportamientos no funcionales que exhibe mientras realiza esas tareas.

* Pruebas realizadas cuando el sistema funciona como un todo.
* Se prueban requerimientos funcionales y requerimientos no funcionales del sistema.
* El entorno de pruebas debe corresponder al entorno de producción tanto como sea posible para reducir los riesgos de incidentes.

#### Objetivos de prueba de sistema

* Reducir el riesgo.
* Verificar que los comportamientos funcionales y no funcionales del sistema son los diseñados y especificados.
* Validar que el sistema está completo y que funcionará como se espera.
* Generar confianza en la calidad del sistema considerado como un todo.
* Encontrar defectos.
* Prevenir la propagación de defectos a niveles de prueba superiores o a producción.

#### Aplicación

Se suele aplicar como base de pruebas para componentes como informes de análisis de riesgo, casos de uso, épicas e historias de usuario o diagramas de estado, entre otras.

Por esto mismo, los objetos de prueba suelen ser aplicaciones, sistemas operativos, sistemas hardware/software, sistema sujeto a prueba o configuraciones del sistema.

#### Enfoque

La prueba de sistema debe centrarse en el comportamiento global y extremo a extremo del sistema en su conjunto, tanto funcional como no funcional. La prueba de sistema debe utilizar las técnicas más apropiadas para los aspectos del sistema que serán probados.

### Prueba de aceptación

#### Definición

La prueba de aceptación, al igual que la prueba de sistema, se centra normalmente en el comportamiento y las capacidades de todo un sistema o producto.

La diferencia es que aquí se evaluará el grado de satisfacción general para su despliegue y uso por parte de los usuarios.

#### Objetivos

* Establecer confianza en la calidad del sistema en su conjunto.
* Validar que el sistema está completo y que funcionará como se espera.
* Verificar que los comportamientos funcionales y no funcionales del sistema sean los especificados.

#### Formas de prueba de aceptación

* Prueba de aceptación de usuario.
* Prueba de aceptación operativa.
* Prueba de aceptación contractual y de regulación.
* Prueba alfa y beta.

#### Prueba de aceptación de usuario

UAT por sus siglas en inglés, se centra normalmente en la validación de la idoneidad para el uso del sistema por parte de los usuarios previstos en un entorno operativo real o simulado.

El objetivo principal es crear confianza en que los usuarios pueden utilizar el sistema para satisfacer sus necesidades, cumplir con los requisitos y realizar los procesos de negocio con la mínima dificultad, coste y riesgo.

#### Prueba de aceptación de operativa

El objetivo principal es generar confianza en que los operadores o administradores del sistema pueden mantener el sistema funcionando correctamente para los usuarios en el entorno operativo, incluso en condiciones excepcionales o difíciles.

#### Prueba de aceptación contractual y normativa

Se realiza en función de los criterios de aceptación del contrato para el desarrollo de software a medida. Los criterios de aceptación deben definirse cuando las partes acuerdan el contrato.

El principal objetivo es crear confianza en que se ha

logrado la conformidad contractual o normativa.

#### Prueba alfa y Beta

Las pruebas alfa y beta suelen ser utilizadas por los desarrolladores de software comercial de distribución masiva que desean obtener retroalimentación de los usuarios, clientes y/u operadores potenciales o existentes antes de que el producto de software sea puesto en el mercado.

##### Prueba alfa

La prueba alfa se realiza en las instalaciones de la organización que desarrolla, no por el equipo de desarrollo, sino por clientes potenciales o existentes, y/u operadores o un equipo de prueba independiente.

##### Prueba beta

La prueba beta es realizada por clientes potenciales o existentes, y/u operadores en sus propias instalaciones.

La prueba beta puede tener lugar después de la prueba alfa, o puede ocurrir sin que se haya realizado ninguna prueba alfa previa.

#### Aplicación

Se suele aplicar como base de pruebas para componentes como procesos de negocio, casos de uso, requisitos de sistema o procedimientos de instalación, entre otras.

Por esto mismo, los objetos de prueba suelen ser sistema sujeto a prueba, formularios, datos de producción, sistema sujeto a prueba o sistemas de recuperación y sitios críticos.

#### Enfoque

La prueba de aceptación es, a menudo, responsabilidad de los clientes, usuarios de negocio, propietarios de producto u operadores de un sistema, y otros implicados también pueden estar involucrados.

La prueba de aceptación se considera, a menudo, como el último nivel de prueba en un ciclo de vida de desarrollo secuencial.

### Prueba de mantenimiento

#### Definición

Una vez que un producto software es lanzado a entornos productivos, comienzan las pruebas de mantenimiento que buscan saber que el software mantiene su calidad en tiempo de vida activa.

También es necesario para preservar o mejorar las características de calidad no funcionales del sistema a lo largo de su vida útil.

#### Activadores para el mantenimiento

Modificación, tales como mejoras planificadas, cambios correctivos y de emergencia, actualizaciones del software y parches para los defectos y las vulnerabilidades.

Migración de una plataforma a otra.

Retirada cuando una aplicación llega al final de su vida útil.

### Relación tipos contra niveles

Con todo lo aprendido en la clase de hoy, podemos empezar a crear ideas de cómo cada tipo y nivel se relacionan en distintas etapas del desarrollo, y cómo se adaptarán acorde a las pruebas y el producto en función.

### Glosario

**Pruebas Funcionales:** pruebas que evalúan las funciones que el sistema debe realizar.

**Pruebas No Funcionales:** la prueba no-funcional prueba que tan bien se comporta un sistema. Se basan en las pruebas de rendimiento y desempeño de un sistema.

**Pruebas de Regresión:** La prueba de regresión consiste en probar un sistema que ha sido analizado previamente para asegurar que no se haya introducido algún tipo de defecto como resultado de cambios realizados

**Prueba Beta:** las pruebas realizadas por usuarios reales en un entorno real.

**Prueba Alfa:** prueba de aceptación interna realizada principalmente por los equipos internos de control de calidad y pruebas del software. La prueba alfa es la última prueba realizada por los equipos de prueba en el sitio de desarrollo después de la prueba de aceptación y antes de lanzar el software para la prueba beta.

**TDD:** desarrollo guiado por pruebas

#### ¿QUIERES SABER MÁS? TE DEJAMOS MATERIAL AMPLIADO DE LA CLASE

* [Documento ISTQB en Español - Página 37 a 57](https://141647fc-7a5b-4407-a84a-78567bb3f5b0.filesusr.com/ugd/acfdb9_743bf3d31dde49578c94d97e5b96b9da.pdf)
* [Pruebas unitarias y Test-Driven Development](https://www.youtube.com/watch?v=YuRdaR6wwWU)
* [Ampliación de Tipos de Pruebas](http://youtube.com/watch?v=3hPCueKXlq4)
* [Ampliación de Niveles de Pruebas](https://www.youtube.com/watch?v=uoiM1BPKfxE)
* [Pruebas de Caja Blanca](https://www.youtube.com/watch?v=GVegCwwfBZ0)
* [Pruebas de Regresión](https://www.youtube.com/watch?v=GVegCwwfBZ0)

#DEMOCRATIZANDOLAEDUCACIÓN