

---

## MÓDULO 3 DE ISQTB FOUNDATION. PRUEBAS ESTÁTICAS

### CODE TESTING

Lorem ipsum dolor sit amet,  
melius commodo splendide  
vix eu, justo graeco sit  
no, mei cu albuscius  
euripidis.



---

### 4.1 CONCEPTOS BÁSICOS DE PRUEBAS ESTÁTICAS

A diferencia de las pruebas dinámicas en las pruebas estáticas **no es necesario ejecutar el código bajo prueba**. El código, la especificación de la arquitectura, otros productos de trabajo del proceso de desarrollo pueden sufrir una **revisión, que no es más que una revisión manual realizada por un probador, o también realizada por una herramienta, análisis estático**.

Una herramienta que realiza análisis estáticos es SonarQube. Los objetivos de las pruebas estáticas son los siguientes:

- Mejorar la calidad.
- Detectar defectos.
- Evaluar legibilidad.
- Evaluar completitud.
- Evaluar corrección.
- Evaluar comprobabilidad.
- Evaluar consistencia.

Las pruebas estáticas se pueden aplicar tanto para la validación como para la verificación.

Los probadores, representantes comerciales y desarrolladores trabajan juntos para que las historias de usuario y los productos de trabajo cumplan con criterios definidos.

La revisión se puede utilizar para que las historias de usuario sean completas, comprensibles y tengan criterios de aceptación comprobables.

El análisis estático puede identificar problemas mucho antes que las pruebas dinámicas y además requiere menos esfuerzo porque no se utilizan casos de prueba.

El análisis estático se utiliza con la integración continua y sirve para:

- Encontrar defectos de código.
- Comprobar el mantenimiento.
- Comprobar la seguridad.

Se suelen utilizar herramientas como **SonarQube** para realizar análisis estático.

#### 4.1.1 Productos de trabajo evaluables mediante pruebas estáticas (Recordar)

Algunos de los productos de trabajo que se pueden examinar con las pruebas estáticas serían:

- Especificación de requisitos.
- Código fuente.
- Planes de prueba.
- Documentos.
- Contratos.
- Modelos.
- Items de la pila (backlog) del producto en un producto con desarrollo ágil.

Realmente cualquier producto de trabajo que se pueda comprender y leer puede ser objeto de una revisión.

Para poder realizar un análisis estático es necesario que el producto de trabajo tenga una estructura que se pueda comprobar (un modelo, código o sintaxis formal).

*Por ejemplo, un código en un lenguaje tendrá una sintaxis que tendrá que ser comprobada por un entorno que valide si está correctamente utilizada.*

*Un modelo entidad-relación para definir una base de datos habrá que comprobar si está correctamente implementado.*

**Un ejemplo de productos que no son apropiados para las pruebas estáticas son aquellos que no son entendidos por los seres humanos o no pueden ser analizados por herramientas como el código ejecutable que no se puede analizar por cuestiones legales.**

### 4.1.2 Valor de las pruebas estáticas (Comprender)

Una de las mejores cosas buenas que tienen las pruebas estáticas es que permiten detectar defectos en las primeras fases del ciclo de vida de desarrollo de software.

También las pruebas estáticas permiten detectar defectos que no se podrían detectar con pruebas dinámicas como los siguientes:

- Código inalcanzable (que nunca se ejecuta).
- Patrones de diseño no implementados correctamente.
- Defectos de productos de trabajo no ejecutables.

Las pruebas estáticas tienen los siguientes beneficios:

- Generan confianza en los productos de trabajo.
- Permiten asegurarse de que los requisitos documentados cumplen las necesidades reales de las partes interesadas.
- Se mejora la comunicación con las partes interesadas, involucrando todo tipo de roles de las partes interesadas en las pruebas estáticas.
- Las revisiones generan menos costes en los proyectos al tardar menos tiempo en resolver defectos en las fases posteriores.
- El análisis estático detecta defectos en el código de manera más eficiente que las pruebas dinámicas lo que lleva a un desarrollo de software con menos esfuerzo y menos defectos en el código.

Hay que comentar que en mi libro **Scrum. Teoría e implementación práctica** en el tema 12 en la sección 12.1 hablo ampliamente de los patrones de diseño.

### 4.1.3 Diferencias entre pruebas estáticas y pruebas dinámicas (Comprender)

Lo primero que hay que saber que las pruebas dinámicas y estáticas son complementarias, pero tienen las siguientes diferencias:

- Las pruebas estáticas y pruebas dinámicas ambas encuentran **defectos**, pero algunos tipos de **defectos** solo se pueden encontrar con pruebas estáticas y otros **defectos** solo con pruebas dinámicas.
- Las pruebas estáticas encuentran defectos directamente en el **objeto de prueba**, pero las pruebas dinámicas encuentran **fallas** asociadas a **defectos** que se encuentran después de realizar un análisis.
- Las pruebas estáticas se encuentran en partes del código que nunca se encontrarían con pruebas dinámicas por ser código que nunca se ejecuta.
- Las pruebas estáticas solo se pueden realizar a productos de trabajo no ejecutables, mientras que las pruebas dinámicas solo se pueden realizar sobre productos de trabajo ejecutables.

- Las pruebas estáticas se pueden utilizar para evaluar características de calidad que no dependen de la ejecución del código (pruebas de mantenimiento, por ejemplo), mientras que las pruebas dinámicas dependen de la ejecución del código como por ejemplo las pruebas de rendimiento.

Los defectos que son más fáciles de encontrar con las pruebas estáticas son:

- Defectos en los requisitos, *por ejemplo, ambigüedades y contradicciones.*
- Defectos de diseño, *por ejemplo, estructura de base de datos ineficientes.*
- Defectos de codificación, *por ejemplo, variables no declaradas, código duplicado, algoritmos con demasiado coste, métodos no descritos adecuadamente.*
- Desviaciones de los estándares, por ejemplo, no utilizar la nomenclatura adecuada en el código.
- Especificaciones de interfaz incorrectas, *por ejemplo, número de parámetros distintos o con tipos de datos diferentes o tipos de valores de retorno de un método incorrecto.*
- Vulnerabilidades de seguridad, *por ejemplo, tamaño de memoria incorrecto que genera un desbordamiento de búfer, filtros incorrectos en servidor que permite ataques XSS.*
- Cobertura de base de pruebas inexacta, *por ejemplo, que no se haya hecho pruebas para los criterios de aceptación y sin embargo la cobertura diga que sí o que no se hayan realizado pruebas unitarias a todos los métodos y caminos y la documentación diga que si se realizaron.*

Con respecto al desbordamiento de búfer como es un concepto de seguridad quizás no se sepa qué es y básicamente cuando se produce un desbordamiento de búfer es cuando un programa que escribe datos en un búfer sobrecarga la capacidad de ese búfer. Es como echar 300 ml de leche en un vaso con 200 ml de capacidad.

Un búfer es un espacio en memoria en el que se almacenan datos de manera temporal, normalmente para un uso concreto como evitar que un programa se quede sin datos en una transferencia irregular o lenta.

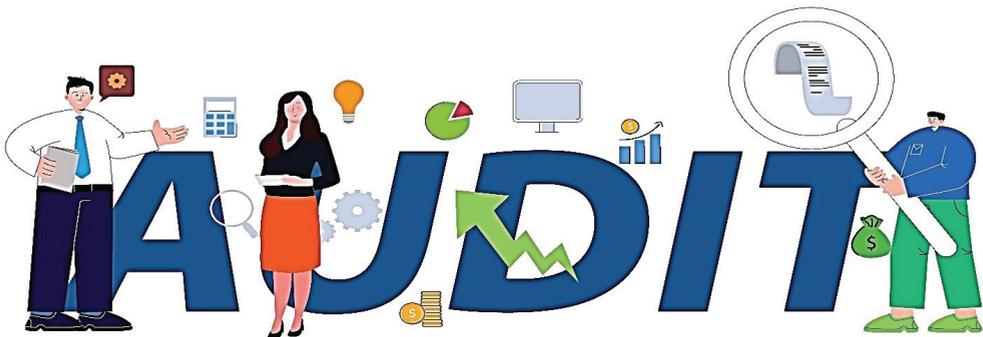
Otro concepto de seguridad es **ataque XSS** o **Cross-site scripting (XSS)** es una vulnerabilidad de seguridad que permite a un atacante inyectar en un sitio web código malicioso del lado del cliente, generalmente enviado por la URL en método de envío **GET** o por formulario en método de envío POST. Este código es ejecutado por las víctimas y permite a los atacantes eludir los controles de acceso y hacerse pasar por usuarios.

*Un ejemplo de pruebas estáticas sería el análisis estático que realiza un producto como **SonarQube** o **SonarCloud** que sería su versión online, la primera hay que instalarla en un servidor o tiene un plugin en entornos de desarrollo como **Eclipse** o **IntelliJ** y nos permite encontrar errores de defectos, mal diseño del código, algoritmos poco eficientes, o fallos de seguridad en el código que permiten por ejemplo un desbordamiento de buffer o también errores a la hora del mantenimiento del código no haciéndolo entendible ni reutilizable.*

*Un ejemplo de pruebas dinámicas que son aquellas en las que hay que ejecutar el código son las herramientas de automatización de pruebas como **Cypress**, **Selenium** o **Katalon Studio**; herramientas que permite automatizar las acciones que se tienen que ejecutar en una aplicación web o como **Katalon** en aplicaciones web, de escritorio o móviles permitiendo ejecutar casos de prueba de manera automatizada y que se suele utilizar mucho en las pruebas de regresión y como se ejecuta el código pues son un ejemplo de pruebas dinámicas, también las pruebas de rendimiento son pruebas dinámicas ya que se necesita ejecutar el código para poder ver los tiempos de respuesta del servidor.*

## 4.2 PROCESO DE RETROALIMENTACIÓN Y REVISIÓN

---



### 4.2.1 Beneficios de la retroalimentación temprana y frecuente de las partes interesadas (Recordar)

La retroalimentación temprana y frecuente con las partes interesadas permite la comunicación temprana de problemas de calidad.

Si hay poca comunicación de las partes interesadas durante el desarrollo del proyecto el producto podría no cumplir la visión de las partes interesadas y entonces podría ocurrir reelaboración del producto con sus costes asociados, plazos incumplidos e incluso el fracaso del proyecto.

Es importante comentarios continuos de las partes interesadas durante todo el ciclo de vida de desarrollo de software para prevenir requisitos mal implementados.

En definitiva, la retroalimentación continua de las partes interesadas ayuda a que los desarrolladores entiendan cuales son las características que ofrecen más valor a las partes interesadas o **stakeholders**.

### 4.2.2 Actividades del proceso de revisión (Comprender)

La norma **ISO/IEC 20246** define un proceso de revisión genérico que se puede adaptar a una situación particular si fuese necesario.

Algunas veces el producto de trabajo es tan grande que se tiene que dividir la revisión en varias revisiones para poder terminarla.

Las actividades del proceso de revisión se pueden ver en la siguiente sección.

#### Planificación

Se define el alcance de la revisión que está formado por:

- El propósito.
- El producto que se revisará.
- Las características de calidad que se evaluarán.
- Las áreas donde enfocarse.
- Los criterios de salida.
- Estándares.
- Plazos de revisión.

## Inicio de revisión

El objetivo es asegurarse que todos los involucrados en la revisión están preparados para comenzarla y tengan:

- Acceso al producto de trabajo bajo revisión.
- Entiendan su rol y responsabilidades.
- Tengan lo necesario para empezar la revisión.

## Revisión individual

Cada revisor realiza una revisión individual del producto de trabajo utilizando dos técnicas, basada en escenarios o en listas de comprobación.

El revisor registra anomalías, recomendaciones y preguntas que necesite realizar.

## Comunicación y análisis

Las anomalías identificadas durante la revisión individual no se saben si son defectos, hay que realizar una reunión de revisión para discutir si es un defecto en función de su propiedad, estado y acciones requeridas y en esa misma reunión los participantes también deciden el nivel de calidad del producto revisado y las acciones de seguimiento que se realizarán.

## Reparación y análisis

Para cada defecto, se debe crear un **informe de defecto** para que se puedan seguir las acciones necesarias para solucionar ese defecto.

Una vez se hayan cumplido los **criterios de salida**, se debe aceptar el producto de trabajo como revisado y se tiene que informar de los productos revisados, su estado a las partes interesadas y al equipo.

### 4.2.3 Roles y responsabilidades en las revisiones (Recordar)

En la siguiente tabla muestro los roles que pueden existir en una revisión y sus funciones.

Director	Autor	Moderador	Escriba	Revisor	Líder de revisión
Decide lo que debe ser revisado y proporciona los recursos necesarios como son personal y tiempo.	Crea y corrige el producto bajo revisión.	También llamado facilitador, gestiona las reuniones de revisión, gestionando el tiempo de la revisión, mediando en las reuniones y asegurando el entorno para la revisión.	También llamado grabador, registra las anomalías encontradas por el revisor, las encontradas en la reunión y las decisiones con respecto a si es un defecto o no.	Es el que realiza las revisiones. Un revisor puede ser alguien del proyecto o un experto externo o no.	Es el responsable de la revisión, el que elige las personas que participaran en la revisión y la hora y lugar donde se realizará la revisión.

#### 4.2.4 Tipos de revisiones (Comprender)

Existen muchos tipos de revisiones, que va desde la informal a la formal. El nivel de la formalidad depende de los siguientes factores:

- El ciclo de vida de desarrollo de software o SDLC que se sigue.
- La madurez del proceso de desarrollo.
- La complejidad del producto que se revisa.
- La importancia del producto a revisar.
- Los requisitos legales.
- Necesidad de una auditoría.

Seleccionar el tipo de revisión correcto es importante para conseguir los objetivos de la revisión, pero el tipo de revisión depende más de factores como los recursos disponibles, necesidades del proyecto, riesgos del producto o dominio del negocio.

Un producto de trabajo puede ser revisado con diferentes tipos de revisión y las revisiones más habituales son las siguientes:

## Revisión informal

Una revisión informal no sigue ningún proceso definido y no es necesario presentar un resultado en un documento formal.

Su objetivo principal es encontrar anomalías.

## Revisión guiada o tutoría

La revisión guiada suele ser dirigida por el autor y tiene el *objetivo principal de evaluar la calidad* y otros objetivos son:

- Educar a los revisores.
- Evaluar la calidad y generar confianza en el producto de trabajo.
- Obtener consenso.
- Generar nuevas ideas.

## Revisión técnica

Una revisión técnica es realizada por revisores cualificados técnicamente y es dirigida por un moderador y sus objetivos principales son *obtener consenso y encontrar una solución a un problema técnico*.

Otros objetivos es evaluar el producto y generar confianza, encontrar anomalías y motivar para que los autores del producto mejoren.

*Un ejemplo de revisión técnica sería algo que me pasó en mi último proyecto, no conseguía encontrar la forma de detectar una ventana modal que aparecía para seleccionar unos de los certificados que se utilizaba para acceder a una web de la administración de justicia y se hizo una reunión con el objetivo de entre los miembros de automatización y 2 experimentados desarrolladores encontrar la solución a ese problema, después de una hora se consiguió, se registró la solución y pudimos seguir con la automatización de los casos de prueba de la aplicación del ministerio de Justicia.*

## Inspección

Es el tipo de revisión más formal por lo tanto sigue todo el proceso de revisión al completo. *El objetivo principal es encontrar el número máximo de anomalías*.

Otros objetivos son evaluar la calidad del producto, generar confianza en el mismo y ayudar a mejorar al autor del producto de trabajo.

Un autor nunca puede tener el rol de líder de revisión o escriba.

### 4.2.5 Factores de éxito para las revisiones (Recordar)

Los factores que determinan el éxito de una revisión son los siguientes, los más importantes están marcados en gris.

- Definir objetivos claros y criterios de salida medibles.
- Elegir el tipo de revisión apropiado.
- Realizar revisiones de trozos pequeños del producto de trabajo para que no se pierda la concentración.
- Proporcionar comentarios a las partes interesadas y a autores.
- Proporcionar tiempo suficiente a los participantes para preparar las revisiones.
- Apoyo de la gerencia.
- Hacer que las revisiones formen parte de la cultura de la empresa.
- Proporcionar una formación adecuada.
- Facilitar reuniones.

### 4.3 GLOSARIO DEL MÓDULO 3

---

- **Anomalía.** Desviación inesperada del comportamiento del sistema.
- **Pruebas dinámicas.** Pruebas en las que se ejecuta el software.
- **Pruebas estáticas.** Pruebas en las que no se ejecuta el software.
- **Revisión formal.** Se sigue un proceso con sus actividades y roles.
- **Revisión informal.** No tiene procesos registrados que haya que seguir.
- **Inspección.** Es un tipo de revisión donde se analiza profundamente al producto.
- **Análisis estático.** Se utiliza con herramientas que analizan el código para encontrar problemas de mantenibilidad y seguridad como puede ser SonarQube.
- **Revisión técnica.** Revisión con el objetivo principal de obtener la solución a un problema técnico.

## 4.4 EXAMEN DEL MÓDULO 3 DE ISTQB



Haz este examen sin mirar las respuestas y sin mirar el contenido del libro porque el objetivo es que aprendas todo lo que necesitas saber para aprobar el certificado **ISTQB Certified Tester Foundation level 4.0** y con estas preguntas sabrás si entendiste todo y te acuerdas; tienes que acertar 5 de 8 para aprobar este examen del módulo 3.

### 1. Pregunta 1 (1 Punto)

¿Cuál de los siguientes no es un beneficio de las pruebas estáticas?

- Tener una gestión de defectos menos costosa debido a la facilidad de detectar defectos más adelante en el SDLC.
- La reparación de los defectos encontrados durante las pruebas estáticas es generalmente mucho menos costosa que la reparación de los defectos encontrados durante las pruebas dinámicas.

- c) Encontrar defectos de codificación que podrían no haberse encontrado realizando únicamente pruebas dinámicas.
- d) Detectar lagunas e inconsistencias en los requisitos.

Seleccione una opción.

## 2. Pregunta 2 (1 Punto)

¿Cuál de los siguientes es un beneficio de la retroalimentación temprana y frecuente?

- a) Mejora el proceso de prueba para futuros proyectos.
- b) Obliga a los clientes a priorizar sus requisitos en función de los riesgos acordados.
- c) Es la única manera de medir la calidad de los cambios.
- d) Ayuda a evitar malentendidos sobre los requisitos.

Seleccione una opción.

## 3. Pregunta 3 (1 Punto)

Las revisiones que se utilizan en su organización tienen los siguientes atributos:

- Existe el papel de un escriba.
- El objetivo principal es evaluar la calidad.
- La reunión es liderada por el autor del producto del trabajo.
- Hay preparación individual.
- Se elabora un informe de revisión.

¿Cuál de los siguientes tipos de revisión es más probable que se utilice?

- a) Revisión informal.
- b) Tutorial.
- c) Revisión técnica.
- d) Inspección.

Seleccione una opción.

**4. Pregunta 4 (1 punto)**

¿Cuál de estas afirmaciones no es un factor que contribuye a que las revisiones sean exitosas?

- a) Los participantes deben dedicar el tiempo adecuado a la revisión.
- b) Dividir los productos de trabajo grandes en partes pequeñas para hacer que el esfuerzo requerido sea menos intenso.
- c) Los participantes deben evitar comportamientos que puedan indicar aburrimiento, exasperación u hostilidad hacia otros participantes.
- d) Las fallas encontradas deben reconocerse, apreciarse y manejarse objetivamente.

Seleccione una opción.

**5. Pregunta 5 (1 Punto)**

¿Cuál de los siguientes es un ejemplo de un defecto que se puede encontrar mediante pruebas estáticas, pero no mediante pruebas dinámicas?

- a) Falta de usabilidad proporcionada a través de la interfaz de usuario.
- b) Código sin camino que lo alcance.
- c) Tiempos de respuesta deficientes para la mayoría de los usuarios esperados.
- d) Funciones requeridas que no están implementadas en el código.

Seleccione una opción.

**6. Pregunta 6 (1 Punto)**

¿Cuál de los siguientes es un beneficio de la retroalimentación temprana y frecuente de las partes interesadas?

- a) Los gerentes son conscientes de qué desarrolladores son menos productivos.
- b) Permite a los gerentes de proyectos priorizar las interacciones con las partes interesadas.
- c) Facilita la comunicación temprana de posibles problemas de calidad.
- d) Los usuarios finales comprenden mejor por qué se retrasa la entrega del producto del trabajo.

Seleccione una opción.

### 7. Pregunta 7 (1 Punto)

Dadas las siguientes descripciones de tareas:

1. Se seleccionan las características de calidad a evaluar y los criterios de salida.
2. Todos tienen acceso al producto del trabajo.
3. Se identifican anomalías en el producto del trabajo.
4. Se discuten las anomalías.

Y las siguientes actividades de revisión:

- A) Revisión individual.
- B) Inicio de la revisión.
- C) Planificación.
- D) Comunicación y análisis.

¿Cuál de las siguientes opciones coincide mejor con las descripciones de tareas y actividades?

- a) 1B, 2C, 3D, 4A.
- b) 1B, 2D, 3C, 4A.
- c) 1C, 2A, 3B, 4D.
- d) 1C, 2B, 3A, 4D.

Seleccione una opción.

### 8. Pregunta 8 (1 Punto)

Dados los siguientes roles en las revisiones:

1. Escriba.
2. Líder de revisión.
3. Facilitador.
4. Gerente.

Y las siguientes responsabilidades en las revisiones:

- A) Garantiza el desarrollo eficaz de las reuniones de revisión y el establecimiento de un entorno de revisión seguro.
- B) Información de revisión de registros, como decisiones y nuevas anomalías encontradas durante la reunión de revisión.
- C) Decide qué se va a revisar y proporciona recursos, como personal y tiempo para la revisión.
- D) Asume la responsabilidad general de la revisión, como organizar cuándo y dónde se llevará a cabo la revisión.

¿Cuál de los siguientes se adapta mejor a los roles y responsabilidades?

- a) 1A, 2B, 3D, 4C.
- b) 1A, 2C, 3B, 4D.
- c) 1B, 2D, 3A, 4C.
- d) 1B, 2D, 3C, 4A.

Seleccione una opción.

## 4.5 RESPUESTAS DEL EXAMEN MÓDULO 3

---

### 1. Pregunta 1 (1 Punto)

Solución

- a) *Es correcto. La gestión de defectos no es menos costosa. Encontrar y reparar defectos más adelante en SDLC es más costoso.*

Mirar sección 4.1.2.

### 2. Pregunta 2 (1 Punto)

Solución

- d) *Es correcto. La retroalimentación temprana y frecuente permite la comunicación temprana de posibles problemas de calidad.*

Mirar sección 4.2.1.