

Testing I

Clase 2



**Certified Tech
Developer**
The Ultimate Degree

Índice

1. Niveles de Prueba
2. Tipos de Pruebas
3. Ejercitación

1 | Niveles de Prueba

Niveles de Prueba

- **Prueba de Componente**

- Objetivos Específicos:
 - Reducir el Riesgo
 - Verificar que los comportamientos funcionales y no funcionales del componente son los esperados.
 - Generar confianza en la calidad del componente.
 - Encontrar defectos en el componente.
 - Prevenir la prolongación de defectos a niveles de prueba superiores.
- Base de prueba
 - Código
 - Modelo de Datos

Niveles de Prueba

- **Prueba de Componente**
 - Objetos de prueba
 - Componentes, unidades o módulos.
 - Códigos y estructuras de datos.
 - Clases
 - Módulos de base de datos.
 - Defectos y fallas características
 - Funcionamiento incorrecto.
 - Problemas de flujo de datos.
 - Código y lógica incorrectos.

Niveles de Prueba

- **Prueba de Integración**

- Objetivos Específicos:
 - 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.
 - Prevenir la propagación de defectos a niveles superiores.
- Base de prueba
 - Diseño de software
 - Casos de Uso
 - Arquitectura

Niveles de Prueba

- **Prueba de Integración**

- Objetos de prueba
 - Subsistemas
 - Bases de Datos
 - Microservicios
- Defectos y fallas características
 - Relacionados a datos.
 - Incompatibilidad de la interfaz
 - Fallos de comunicación entre componentes.

Niveles de Prueba

- **Prueba de Sistema**

- **Objetivos Específicos:**
 - Verificar que los comportamientos funcionales y no funcionales del sistema son los diseñados y especificados.
 - Validar que el sistema está completo.
 - Considerar el sistema como un todo.
 - Encontrar defectos.
 - Prevenir la prolongación de defectos en producción.
- **Base de prueba**
 - Informes de análisis de prueba.
 - Casos de uso.
 - Manuales del sistema y del usuario.

Niveles de Prueba

- **Prueba de Sistema**

- Objetos de prueba
 - Aplicaciones, sistemas operativos.
- Defectos y fallas características
 - Cálculos incorrectos.
 - Fallas en tareas funcionales de extremo a extremo.
 - Falla en entornos productivos.
 - Fallo del sistema para funcionar como describen los manuales.

Niveles de Prueba

- **Prueba de Aceptación**

- **Objetivos Específicos:**
 - 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 son los especificados.
- **Base de prueba**
 - Procesos de Negocio
 - Normativas, contratos legales y estándares.
 - Documentación del sistema o del usuario.

Niveles de Prueba

- **Prueba de Aceptación**

- Objetos de prueba
 - Sistema bajo prueba
 - Sistemas de recuperación y sitios críticos.
- Defectos y fallas características
 - Los flujos del sistema no cumplen con los requisitos de negocio o de usuario.
 - El sistema no satisface los requisitos contractuales o reglamentarios.
 - Vulnerabilidades de seguridad o eficiencia de rendimiento inadecuada.

2 | Tipos de Pruebas

Prueba Funcional

- La prueba funcional de un sistema incluye pruebas que evalúan las funciones que el sistema debe realizar. Las funciones describen qué hace el sistema.
- La prueba funcional observa el comportamiento del software.



Prueba No Funcional

- La prueba no funcional prueba “cómo de bien” se comporta el sistema.
- El diseño y la ejecución de la prueba no funcional pueden implicar competencias o conocimientos especiales, como el conocimiento de las debilidades inherentes a un diseño o tecnología (por ejemplo, vulnerabilidades de seguridad asociadas con determinados lenguajes de programación).



3 | Ejercitación

Ejercicio 1

¿Cual de las siguientes afirmaciones describe MEJOR las pruebas NO funcionales?

- a- Las pruebas no funcionales son el proceso de prueba de un sistema integrado para verificar que cumple con los requisitos especificados.
- b- Las pruebas no funcionales son el proceso de pruebas para determinar la conformidad del sistema con los estándares de codificación.
- c- Las pruebas no funcionales son probar sin hacer referencia a la estructura interna del sistema.
- d- Las pruebas no funcionales son probar características del sistema, tales como la usabilidad, fiabilidad o mantenibilidad.

Ejercicio 2

¿Qué es importante hacer cuando se trabaja con modelos de desarrollo de software?

- a- Si fuera necesario, adaptar los modelos a las características del proyecto y del producto.
- b- Elegir el modelo en cascada, porque es el modelo más consolidado.
- c- Comience con el modelo-V y, a continuación, pasar al modelo iterativo o al incremental.
- d- Cambiar la organización que se adapte al modelo, no al revés.

Ejercicio 3

¿Cuál de las siguientes afirmaciones comparando las pruebas de componentes y las pruebas de sistema es VERDADERA?

a- Las pruebas de componentes verifican la funcionalidad de los módulos de software, objetos del programa y clases que pueden ser probadas por separado, mientras que las pruebas de sistema verifican las interfaces entre los componentes y las interacciones entre las diferentes partes del sistema.

b- Los casos de prueba para las pruebas de componentes, en general, se obtienen a partir de las especificaciones de componente, especificaciones de diseño o modelos de datos, mientras que los casos de prueba para pruebas de sistema se suelen obtener a partir de especificaciones de requisitos, especificaciones funcionales, o casos de uso.

Ejercicio 4 - Elaboración de mapa conceptual grupal

Elaborar un mapa conceptual relacionando todos los conceptos que se vieron en la clase virtual (tipos y niveles de prueba). Si lo consideran necesario pueden agregar algunos conceptos vistos en la primer clase.

Para la elaboración del mapa conceptual les pediremos que utilicen la herramienta <https://app.diagrams.net/>

Se puede utilizar de forma colaborativa y se guarda en la nube de google drive.

Los mapas elaborados serán utilizados por el profe en la clase de cierre de la semana.

DigitalHouse>
Coding School