

Definiciones y concepto:.....	2
Enfoque de Testing: definir que es un enfoque de testing y explicar brevemente los diferentes enfoques que existen.....	2
Nivel de testing: definir que es un nivel de testing y explicar brevemente los diferentes niveles que existen.....	2
Tipo de testing: definir que es un tipo de testing y explicar brevemente los diferentes tipos que existen.....	2
Ejemplos.....	3
Nivel de Testing: .....	4
2. A/B Testing:.....	4
3. diferencias entre Esquema, Nivel y Tipo de Testing.....	5

## Trabajo practico 2 fundamentos de control de calidad

**Nombre y Apellido:** Guanco Yonathan Jesus

Definiciones y concepto:

Enfoque de Testing: definir que es un enfoque de testing y explicar brevemente los diferentes enfoques que existen.

Un enfoque de pruebas se refiere a la implementación de la estrategia de pruebas definida para un proyecto específico<sup>1</sup>. En otras palabras, es la manera en que se planifica y ejecuta el proceso de pruebas para evaluar la calidad y el funcionamiento de un producto o aplicación de software.

Nivel de testing: definir que es un nivel de testing y explicar brevemente los diferentes niveles que existen.

Los **niveles de pruebas de software** se definen como grupos de actividades de prueba que se organizan y gestionan en conjunto, y se realizan según el nivel de desarrollo en que se encuentra el producto

Los **niveles de pruebas de software** se definen como grupos de actividades de prueba que se organizan y gestionan en conjunto, y se realizan según el nivel de desarrollo en que se encuentra el producto. Estos niveles son:

1. **Pruebas de Componente o Unitarias:** En este nivel de pruebas, se evalúan los **componentes individuales** que se pueden probar por separado. Estas pruebas se centran en verificar que cada módulo o unidad funcione correctamente y cumpla con sus especificaciones<sup>1</sup>.
2. **Pruebas de Integración:** Aquí se evalúa la **interacción entre los componentes** del sistema. El objetivo es detectar problemas que puedan surgir al combinar diferentes módulos. Se verifica que las interfaces entre los componentes funcionen adecuadamente y que la integración no cause errores<sup>2</sup>.
3. **Pruebas de Sistema:** En este nivel, se evalúa el **sistema completo** como una entidad. Se verifican las funcionalidades, la comunicación entre módulos y la interacción con otros sistemas o componentes externos. El objetivo es asegurar que el sistema cumpla con los requisitos y funcione correctamente en su conjunto<sup>2</sup>.
4. **Pruebas de Aceptación:** Estas pruebas se realizan para validar que el sistema cumple con los **requisitos del usuario** y está listo para su entrega. Los usuarios finales o representantes del cliente participan en estas pruebas para verificar que el sistema se ajusta a sus necesidades y expectativas<sup>2</sup>.

En resumen, los niveles de pruebas permiten analizar con mayor detalle los riesgos que un componente, aplicación o sistema puede tener, logrando un proceso de pruebas más eficaz. Además, existen tipos de pruebas que se relacionan con estos niveles, como las pruebas funcionales y no funcionales, que miden qué hace el sistema y cómo lo hace, respectivamente<sup>2</sup>

Tipo de testing: definir que es un tipo de testing y explicar brevemente los diferentes tipos que existen.

**El testing** es una parte fundamental del proceso de desarrollo de software. Consiste en evaluar y verificar que el software funcione correctamente y cumpla con los requisitos establecidos. Existen

varios tipos de pruebas, cada uno con su enfoque y propósito específico. A continuación, te presento algunos de los más comunes:

5. **Pruebas Unitarias:**

- Son de **bajo nivel** y se realizan cerca de la fuente de la aplicación.
- Se centran en probar **métodos y funciones individuales** de las clases, componentes o módulos utilizados por el software.
- [Suelen ser baratas de automatizar y se ejecutan rápidamente mediante un servidor de integración continua<sup>1</sup>.](#)

6. **Pruebas de Integración:**

- Verifican que los **módulos o servicios** utilizados por la aplicación funcionen bien en conjunto.
- Evalúan la **interacción entre diferentes partes** del sistema.
- [Detectan problemas de comunicación o compatibilidad entre componentes<sup>1</sup>.](#)

7. **Pruebas de Sistema:**

- Evalúan el **sistema completo** como una unidad.
- Verifican que todos los componentes se integren correctamente y cumplan con los requisitos funcionales y no funcionales.
- [Pueden incluir pruebas de rendimiento, seguridad y usabilidad<sup>2</sup>.](#)

8. **Pruebas de Aceptación:**

- Se realizan para validar que el software cumple con los **requisitos del cliente**.
- Pueden ser **manuales o automatizadas**.
- [A menudo se basan en escenarios de uso real y casos de prueba específicos<sup>1</sup>.](#)

9. **Pruebas de Carga y Estrés:**

- **Pruebas de Carga** evalúan el rendimiento del software bajo **cargas máximas**.
- [Pruebas de Estrés someten al sistema a condiciones extremas para verificar su estabilidad y resistencia<sup>3</sup>.](#)

10. **Pruebas de Seguridad:**

- Evalúan la **resistencia del software a ataques** y vulnerabilidades.
- [Buscan identificar posibles brechas de seguridad y proteger los datos y la integridad del sistema<sup>4</sup>.](#)

Pruebas de Usabilidad:

- Se centran en la **experiencia del usuario**.
- [Evalúan la facilidad de uso, la navegación y la intuitividad del software<sup>1</sup>.](#)

Recuerda que la elección del tipo de prueba depende del contexto del proyecto y los objetivos específicos. Cada tipo de prueba contribuye a garantizar la calidad y confiabilidad del software final.

## Ejemplos

Para proporcionar ejemplos de cada enfoque de testing, es importante considerar tanto el nivel como el tipo de testing. A continuación, se presentan tres ejemplos de aplicación práctica para cada enfoque:

## Nivel de Testing:

### 1. Pruebas Unitarias:

- **Ejemplo**: En el desarrollo de una aplicación web, se realizan pruebas unitarias en una función específica que calcula el total de una compra en un carrito de compras virtual.

### 2. Pruebas de Integración:

- **Ejemplo**: En un sistema de gestión de inventario, se llevan a cabo pruebas de integración para verificar la comunicación entre el módulo de ventas y el módulo de almacén.

### 3. Pruebas Funcionales:

- **Ejemplo**: En una aplicación de banca en línea, se realizan pruebas funcionales para asegurar que los usuarios puedan realizar transferencias de fondos de manera segura y eficiente.

## Tipo de Testing:

### 1. Pruebas Exploratorias:

- **Ejemplo**: Un equipo de testing realiza pruebas exploratorias en una nueva función de chat en una plataforma de redes sociales para identificar posibles problemas de usabilidad y rendimiento.

### 2. A/B Testing:

- **Ejemplo**: Una empresa de comercio electrónico realiza un A/B testing en la página de inicio para comparar dos versiones diferentes y determinar cuál genera más conversiones de ventas.

### 3. **\*\*Testing Multipágina\*\***:

- **\*\*Ejemplo\*\***: Una compañía de software realiza un testing multipágina en su sitio web para evaluar cómo diferentes elementos visuales afectan la tasa de clics en los enlaces de descarga de su producto.

Estos ejemplos ilustran la aplicación práctica de diferentes enfoques de testing en distintos niveles y tipos de pruebas, demostrando su importancia en el aseguramiento de la calidad del software.

Citaciones:

[1] <https://devexpert.io/testing-nivel-2-que-hace-que-un-test-sea-un-test/>

[2] <https://vwo.com/es/ab-testing/>

[3] <https://qalified.com/es/blog/exploratory-testing/>

[4] <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>

[5] <https://es.linkedin.com/advice/3/how-can-you-determine-appropriate-level-testing-your-ztanf?lang=es>

#### 3. diferencias entre Esquema, Nivel y Tipo de Testing

concepto	Objetivos	ejemplos	Nivel de testing	Tipo de testing
Pruebas unitarias	Verificar el correcto funcionamiento de unidades	Función de cálculo de total de	Unidad	Caja blanca

	individuales de código			
<b>Pruebas de integración</b>	<b>Validar la interacción entre diferentes módulos o componente</b>	<b>Comunicación entre módulos de ventas y almacén</b>	<b>Integración</b>	<b>Caja negra</b>
<b>Pruebas funcionales</b>	<b>Asegurar que el software cumpla con los requisitos funcionales específicos</b>	<b>Transferencias de fondos en banca en línea</b>	<b>sistema</b>	<b>Caja negra</b>
<b>Pruebas exploratorias</b>	<b>Identificar problema de usabilidad, rendimiento y funcionalidad</b>	<b>Función de chat en redes sociales</b>	<b>sistema</b>	<b>Caja negra</b>
<b>A/B Testing</b>	<b>Comparar dos versiones y determinar cual tiene mejor rendimiento</b>	<b>Página de inicio de comercio electrónico</b>	<b>Sistema</b>	<b>Caja negra</b>
<b>Testing multipágina</b>	<b>Evaluar el impacto de elementos visuales en el comportamiento del usuario</b>	<b>Sitio web de descarga de software</b>	<b>Sistema</b>	<b>Caja negra</b>
<b>Pruebas de aceptación</b>	<b>Confirmar que el software cumple con los criterios de</b>	<b>Aplicación de gestión de recursos</b>	<b>sistema</b>	<b>Caja negra</b>

	<b>aceptación del cliente</b>			
<b>Pruebas de regresión</b>	Verificar que los cambios no afecten funcionalidad es existentes	<b>Actualización de sistema operativo móvil</b>	<b>Sistema</b>	<b>Caja negra y caja blanca</b>
<b>Pruebas de rendimiento</b>	<b>Evaluar la velocidad estabilidad y estabilidad bajo diferentes cargas</b>	<b>Aplicación de streaming de video</b>	<b>sistema</b>	<b>Caja negra</b>