

Universidad Mariano Galvez de Guatemala

Ingeniería En Sistemas de Información

Sede Chiquimulilla

Curso: Aseguramiento de Calidad de Software

Docente: Carmelo Mayen



#### Tarea 4: Tipos de prueba

#### **Estudiante:**

Armando Sierra González

Carné: 1790-21-18438

Chiquimulilla Agosto 2025

## Introducción

El presente trabajo tiene como objetivo investigar herramientas esenciales para la realización de pruebas de software, centrándose en tres tipos fundamentales: pruebas funcionales, pruebas de rendimiento y pruebas de seguridad. Las pruebas de software son un pilar en el desarrollo de aplicaciones, ya que permiten garantizar que el sistema cumpla con los requisitos establecidos, sea confiable, seguro y eficiente. Cada tipo de prueba abordado en este documento desempeña un papel crucial en el ciclo de vida del software: las pruebas funcionales verifican el cumplimiento de la lógica y funciones del sistema, las pruebas de rendimiento evalúan la capacidad de respuesta y estabilidad bajo diversas condiciones de carga, y las pruebas de seguridad identifican vulnerabilidades que podrían ser explotadas por atacantes.

## 1. Pruebas Funcionales

### Definición:

Son aquellas que validan que el software cumple con las funcionalidades especificadas en los requisitos.

### Herramientas:

- **Selenium:** Herramienta de automatización de pruebas para aplicaciones web.
  - **Características:** Open-source, soporta múltiples navegadores, permite pruebas en diversos lenguajes de programación.
  - **Ventajas:** Gratuita, gran comunidad, integración con CI/CD.
  - **Desventajas:** Configuración inicial compleja, requiere conocimientos técnicos.
  - **Aplicación:** Automatización de pruebas de interfaces gráficas en entornos web.
- **TestComplete:** Herramienta comercial para pruebas automatizadas.
  - **Características:** Interfaz amigable, soporta pruebas web, móviles y de escritorio.
  - **Ventajas:** Fácil de usar, soporte técnico.
  - **Desventajas:** Costosa, dependiente de licencia.
  - **Aplicación:** Automatización de pruebas en proyectos con presupuesto para herramientas comerciales.

## 2. Pruebas de Rendimiento

### Definición:

Evalúan la capacidad del software para operar eficientemente bajo ciertas cargas de trabajo.

### Herramientas:

- **JMeter:** Herramienta open-source para pruebas de carga y rendimiento.

- **Características:** Soporta múltiples protocolos, fácil integración con CI/CD.
- **Ventajas:** Gratuita, extensible.
- **Desventajas:** Curva de aprendizaje media.
- **Aplicación:** Simulación de cargas de usuarios en entornos de prueba.
- **LoadRunner:** Herramienta comercial para pruebas de rendimiento.
  - **Características:** Potente motor de simulación de usuarios.
  - **Ventajas:** Alta precisión, reportes detallados.
  - **Desventajas:** Costosa, requiere licencia.
  - **Aplicación:** Evaluaciones de rendimiento en entornos empresariales.

### 3. Pruebas de Seguridad

#### Definición:

Analizan la aplicación para detectar vulnerabilidades que puedan comprometer la seguridad.

#### Herramientas:

- **OWASP ZAP:** Herramienta gratuita para análisis de seguridad de aplicaciones web.
  - **Características:** Escaneo pasivo y activo, integración con CI/CD.
  - **Ventajas:** Open-source, gran comunidad.
  - **Desventajas:** Menos avanzada que herramientas comerciales.
  - **Aplicación:** Evaluar seguridad de aplicaciones web.
- **Burp Suite:** Herramienta comercial de pruebas de penetración.
  - **Características:** Escaneo avanzado, proxy de interceptación.
  - **Ventajas:** Potente, muy usada por profesionales.
  - **Desventajas:** Costosa.

- **Aplicación:** Pruebas de seguridad exhaustivas en entornos críticos.

## Conclusiones

Las pruebas funcionales, de rendimiento y de seguridad son esenciales para garantizar la calidad y fiabilidad del software.

La elección de la herramienta adecuada depende del presupuesto, los conocimientos del equipo y las necesidades específicas del proyecto.

Integrar estas pruebas de manera sistemática en el ciclo de desarrollo es clave para prevenir fallos, optimizar el rendimiento y proteger el sistema contra amenazas.

## Recomendaciones

Para seleccionar la herramienta adecuada, se recomienda:

- Evaluar el presupuesto disponible.
- Considerar el nivel técnico del equipo.
- Analizar las características específicas que se necesitan.
- Integrar las pruebas en un flujo CI/CD para mayor eficiencia.

## Bibliografia

- SeleniumHQ. (2025). *The Selenium Browser Automation Project*. Recuperado de <https://www.selenium.dev>
- Apache Software Foundation. (2025). *Apache JMeter*. Recuperado de <https://jmeter.apache.org>
- OWASP Foundation. (2025). *Zed Attack Proxy (ZAP)*. Recuperado de <https://www.zaproxy.org>
- Micro Focus / OpenText. (2025). *LoadRunner*. Recuperado de <https://software.microfocus.com/en-us/software/loadrunner>
- PortSwigger Ltd. (2025). *Burp Suite*. Recuperado de <https://portswigger.net/burp>