

Curso de **TESTING AUTOMATION AVANZADO con SELENIUM y JMETER**

OBJETIVO DEL CURSO

Capacitar a los participantes en el diseño e implementación de frameworks de automatización de pruebas avanzados y en la integración fluida de scripts funcionales y de rendimiento dentro de un ciclo de Integración Continua (CI/CD) utilizando herramientas modernas basadas en la nube. Al finalizar, el participante será capaz de desplegar soluciones de automatización escalables y profesionales adaptadas a entornos de desarrollo ágiles y remotos.

ESTRUCTURA DETALLADA DE LAS SESIONES

1. Arquitectura y Frameworks Avanzados en Selenium

- ✓ Patrón de Diseño Page Object Model (POM) Avanzado: Uso profundo e implementación robusta con PageFactory.
- ✓ Data-Driven Testing (DDT): Centralización y lectura/escritura de datos dinámicos mediante archivos complejos (Excel con Apache POI / JSON) para independizar el código de los datos de prueba.
- ✓ Estrategias avanzadas de sincronización: Control de esperas explícitas y personalizadas en elementos dinámicos.
- ✓ Manejo avanzado de interfaces: Interacción con Shadow DOM, iFrames anidados y elementos web altamente inestables.

2. Behavior-Driven Development (BDD) con Cucumber

- ✓ Introducción al enfoque BDD: Filosofía, ventajas y alineación entre equipos de negocio y desarrollo.
- ✓ Sintaxis Gherkin: Redacción profesional de escenarios de prueba utilizando palabras clave (Given, When, Then, And, But).
- ✓ Integración de Cucumber con Selenium WebDriver y Java.
- ✓ Desarrollo de Step Definitions robustos, expresiones regulares y parametrización de escenarios desde archivos de características (.feature).

3. Escalabilidad: Ejecución en la Nube y Paralelismo

- ✓ Introducción a la ejecución distribuida: Conceptos clave de Selenium Grid (Hub y Nodos).
- ✓ Ejecución en Paralelo: Configuración avanzada de TestNG para lanzar pruebas simultáneas en múltiples navegadores y optimizar tiempos de feedback.
- ✓ Automatización en la Nube: Integración y ejecución de scripts utilizando plataformas cloud líderes (como BrowserStack o SauceLabs) mediante cuentas de evaluación, simulando entornos reales de trabajo remoto.

- ✓ Contenedores (Opcional/Demostrativo): Introducción al uso de imágenes ligeras de Docker para pruebas locales aisladas.

4. JMeter Avanzado: Scripting Dinámico y Casos Complejos

- ✓ Scripting con JSR223 y Groovy: Extensión de las capacidades nativas de JMeter para manejar lógicas complejas, encriptación y manipulación avanzada de variables.
- ✓ Correlación Avanzada de Datos: Técnicas para extraer, almacenar y reutilizar tokens dinámicos de seguridad (OAuth, JWT, tokens de sesión) mediante Regular Expression Extractors y JSON Post-Processors.
- ✓ Simulación de Carga Realista: Configuración avanzada de controladores lógicos (Throughput Controller, Switch Controller) y emulación de restricciones de ancho de banda.

5. JMeter: Pruebas Distribuidas y Monitoreo en Grafana Cloud

- ✓ Arquitectura JMeter Distribuido: Configuración lógica del modelo Master-Slave para la generación de alta concurrencia sin saturar el equipo local.
- ✓ Monitoreo de Infraestructura: Conceptos fundamentales para evaluar el consumo de recursos (CPU, Memoria, I/O) en servidores bajo estrés.
- ✓ Integración con Grafana Cloud: Configuración de la capa gratuita de Grafana e InfluxDB para el envío de métricas en tiempo real, reemplazando gráficos estáticos por dashboards profesionales interactivos en la nube.

6. Integración Continua (CI/CD) y Reportes Profesionales

- ✓ Ejecución Non-GUI y CLI: Automatización de suites completas desde la consola de comandos mediante Maven/Gradle y modo nativo de JMeter.
- ✓ Diseño de Pipelines en la Nube: Configuración práctica de un flujo de trabajo (Workflow) automatizado en GitHub Actions o GitLab CI para activar ejecuciones automáticas tras cambios de código.
- ✓ Reportes Visuales de Alto Impacto: Implementación e integración de Allure Reports para la generación de tableros de resultados estéticos, detallados e históricos listos para presentar a stakeholders.

Conocimientos Previos Recomendados

Para garantizar el máximo aprovechamiento de este nivel avanzado, se sugiere que los estudiantes cuenten con:

- Comprensión sólida de la automatización funcional básica con Selenium WebDriver y manejo de localizadores (ID, CSS, XPath).
- Conocimientos básicos del diseño de pruebas de carga iniciales en Apache JMeter (peticiones HTTP, listeners esenciales).
- Familiaridad intermedia con lenguaje de programación Java (POO, estructuras de control, colecciones) y manejo de un IDE.

- Nociones fundamentales de HTML, CSS, selectores y bases de datos relacionales (consultas SQL básicas).
- Manejo básico de comandos Git y uso de repositorios remotos GitHub. (Iniciar un proyecto, realizar commits, subir cambios y clonar repositorios)
- Conocimiento básico de uso de Docker. (Descargar imágenes, crear y arrancar contenedores)