Módulo 4: Automatización de pruebas

Pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

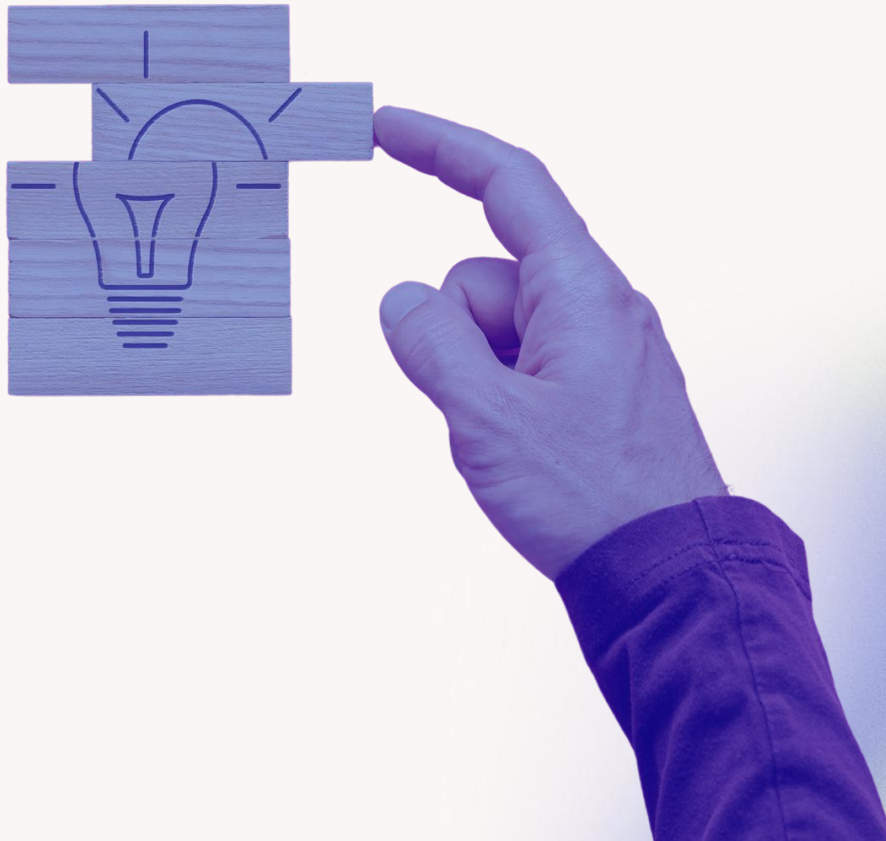
Módulo 4

- Testing en un entorno ágil
- El concepto de regresividad en las pruebas
- Pruebas unitarias
- Pruebas unitarias de servicios web
- ***Pruebas de rendimiento***
- Pruebas funcionales

aspasia
LA FORMACIÓN DE TU FUTURO

Objetivos

Comprender los fundamentos de las pruebas de rendimiento, su aplicación con Apache JMeter y cómo integrarlas en entornos de desarrollo y CI/CD para asegurar la estabilidad y eficiencia del sistema.



aspasia
LA FORMACIÓN DE TU FUTURO

¿Qué impacto tiene el rendimiento de una aplicación en la experiencia del usuario y cómo podrías evaluarlo?

aspasia
LA FORMACIÓN DE TU FUTURO

Fundamentos de Pruebas de Rendimiento

¿Qué son las pruebas de rendimiento y por qué son importantes?

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Las **pruebas de rendimiento** permiten evaluar **cómo responde un sistema bajo determinadas condiciones de uso**, simulando escenarios del mundo real. Se enfocan en asegurar la escalabilidad, velocidad y estabilidad.

✅ **Importancia clave:**

- Detectan cuellos de botella antes de la producción.
- Validan si el sistema puede soportar la carga esperada.
- Previenen incidentes críticos en ambientes reales.
- Aseguran una buena experiencia de usuario bajo carga.

📖 **Ejemplo práctico:**

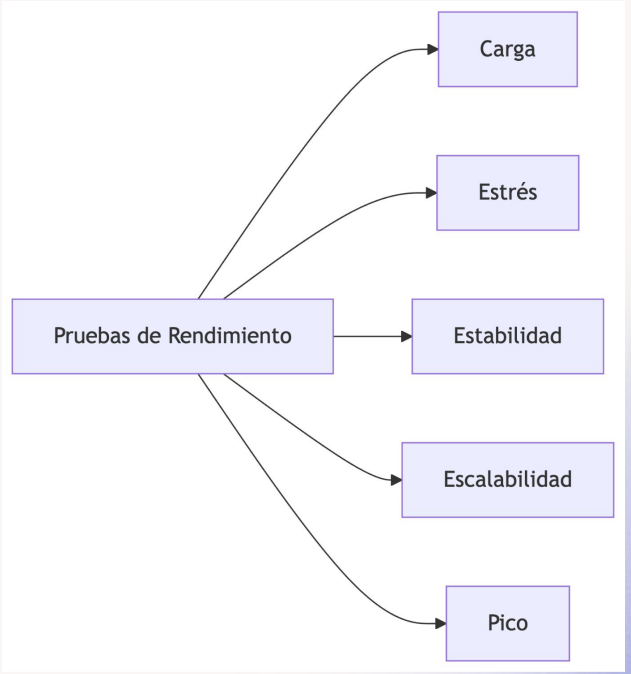
Antes de lanzar un nuevo endpoint /pago, se ejecutan pruebas para verificar que el sistema pueda atender **1000 solicitudes por minuto sin caídas**.

Tipos de pruebas de rendimiento y sus objetivos

aspasia
LA FORMACIÓN DE TU FUTURO

✅ Clasificación común:

Tipo de prueba	Objetivo principal
Carga (Load)	Medir desempeño bajo carga esperada.
Estrés (Stress)	Descubrir el punto de ruptura del sistema.
Estabilidad	Validar comportamiento durante períodos prolongados.
Escalabilidad	Verificar crecimiento progresivo de la carga.
Pico (Spike)	Evaluar respuesta ante aumentos repentinos.



Pruebas de carga: cómo medir el comportamiento del sistema

aspasia
LA FORMACIÓN DE TU FUTURO

✅ **Parámetros comunes:**

- Nº de usuarios concurrentes.
- Nº de peticiones por segundo.
- Tiempo de respuesta promedio.

📖 **Ejemplo conceptual con API REST:**

Simular 500 usuarios accediendo simultáneamente a /login durante 5 minutos.

📊 **Resultado esperado:**

- Tiempo promedio < 1s.
- Tasa de error < 1%.

Pruebas de estrés: identificación de límites de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Buscan encontrar **el punto en el que el sistema deja de responder adecuadamente**, es decir, su límite.

✅ **Qué se observa:**

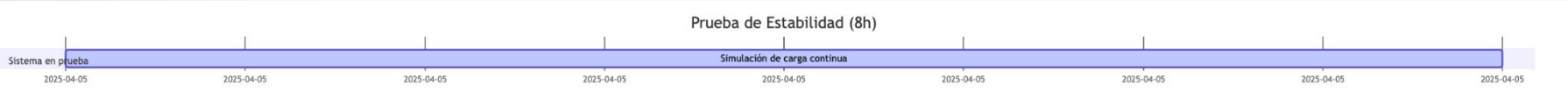
- Cuándo colapsa el sistema.
- Comportamiento post-falla.
- Tiempo de recuperación.

📖 **Ejemplo con JMeter:** Simular carga creciente (100, 200, 500, 1000 usuarios) hasta que el sistema arroje errores 500.

Pruebas de estabilidad: evaluación del sistema a largo plazo

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Evalúan cómo se comporta un sistema bajo carga **sostenida durante horas o días**.



✅ **Busca identificar:**

- Fugas de memoria.
- Degradación de rendimiento.
- Problemas de conexión persistente.

📖 **Caso común:**

Ejecutar solicitudes a /ordenes por 8 horas simulando 50 usuarios activos por minuto.

Factores clave en la ejecución de pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

✅ **Consideraciones importantes:**

- Escenarios realistas (mismo flujo del usuario).
- Datos variables (inputs dinámicos).
- Ambiente controlado (similar a producción).
- Monitoreo activo (CPU, RAM, I/O).

📖 **Recomendación:**

Evita probar en entornos locales. Usa staging o infraestructura en la nube aislada.

Métricas esenciales en pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Las métricas cuantifican el comportamiento del sistema.

✅ **Métricas clave:**

Métrica	Significado
Latencia	Tiempo entre la petición y la primera respuesta
Throughput	Cantidad de peticiones completadas por segundo
Tasa de error	% de respuestas con códigos 4xx o 5xx
Tiempo medio	Promedio de duración de las respuestas
Concurrencia	Nº de usuarios activos al mismo tiempo

📖 **Ejemplo de salida en JMeter:**

```
Throughput: 920 req/s
Avg response time: 780ms
Error %: 0.21%
```

Herramientas populares para pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

✅ Herramientas destacadas:

Herramienta	Lenguaje/Plataforma	Tipo de prueba
JMeter	Java / GUI y CLI	REST/SOAP, carga
Gatling	Scala / Script	APIs, Web, Socket
Locust	Python / Código	Carga distribuida
k6	JS / CLI	Pruebas modernas
Artillery	Node.js / YAML	Microservicios

📖 Ejemplo básico en k6:

```
import http from 'k6/http';

export default function () {
  http.get('https://api.ejemplo.com/login');
}
```

Integración de pruebas de rendimiento en el ciclo de desarrollo

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Es esencial automatizar la ejecución de estas pruebas en el pipeline.

✅ **Integración recomendada:**

- Etapa en CI para ejecución de carga básica.
- Etapas avanzadas en entornos de staging/preproducción.
- Reglas para bloquear despliegues si la latencia es > 1s o errores > 2%.

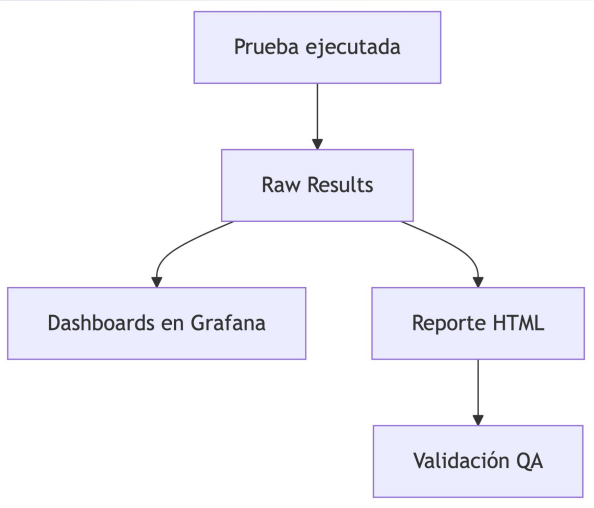
📖 **Ejemplo en GitHub Actions:**

```
- name: Ejecutar prueba de rendimiento
  run: k6 run test-carga.js
```

Reportes y análisis de resultados en pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Tras cada ejecución, se deben analizar los resultados de forma estructurada.



✅ **Reportes típicos incluyen:**

- Curvas de carga vs latencia.
- Histogramas de distribución de respuesta.
- Logs de errores por endpoint.
- Comparativas históricas (antes/después de cambios).

📖 **Recomendación final:**

Guarda todos los reportes en un repositorio de evidencia para auditoría y comparativas.

aspasia
LA FORMACIÓN DE TU FUTURO

¿Qué tipo de prueba de rendimiento utilizarías para evaluar la estabilidad de una aplicación en producción?

aspasia
LA FORMACIÓN DE TU FUTURO

Implementación de Pruebas de Rendimiento con JMeter

¿Qué es JMeter y dónde obtenerlo?

aspasia
LA FORMACIÓN DE TU FUTURO

📌 **Apache JMeter** es una herramienta open-source basada en Java, diseñada para ejecutar **pruebas de rendimiento, carga y estrés** sobre aplicaciones web, servicios REST/SOAP, bases de datos, y más.

✅ **Características clave:**

- Soporta múltiples protocolos (HTTP, JDBC, FTP, etc.).
- Permite simular usuarios concurrentes.
- Genera reportes gráficos y estadísticas detalladas.
- Integrable con CI/CD (Jenkins, GitHub Actions, etc.).

🔗 **Descarga oficial:**
https://jmeter.apache.org/download_jmeter.cgi

Creación y configuración de un Thread Group en JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

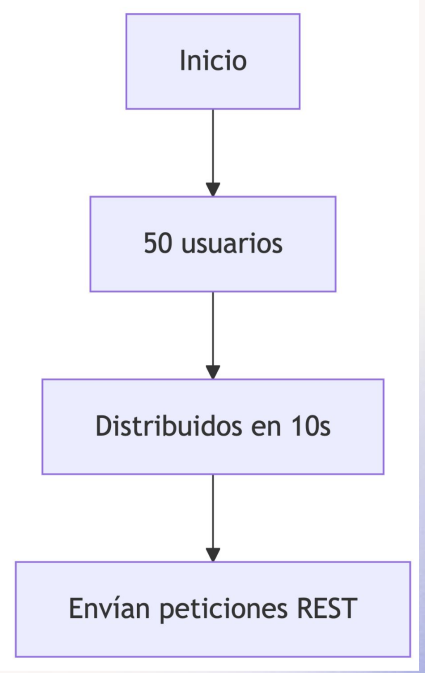
📌 El **Thread Group** simula usuarios virtuales.

✅ **Parámetros principales:**

- **Number of Threads (users):** cuántos usuarios simultáneos.
- **Ramp-Up Period:** tiempo para alcanzar todos los usuarios.
- **Loop Count:** cuántas veces se repetirá cada petición.

📖 **Ejemplo de configuración:**

- 50 usuarios
- Ramp-up: 10 segundos
- Loop: 5 veces



Uso de JMeter Listeners para analizar resultados

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Los **Listeners** permiten visualizar y exportar resultados.

✅ **Listeners comunes:**

- View Results Tree (detallado por solicitud).
- Summary Report (tiempos promedios, errores, throughput).
- Graph Results.
- Backend Listener (para integración con InfluxDB/Grafana).

📖 **Consejo:** No uses muchos listeners activos durante pruebas intensas. Úsalos para debug, y luego analiza resultados desde archivos .jtl.

Creación y ejecución de pruebas de carga con JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Simularemos una carga sobre un endpoint REST (ej. /api/login).

✅ **Pasos básicos:**

1. Añadir un Thread Group.
2. Agregar un **HTTP Request Sampler**.
3. Configurar método (GET/POST), host, path, parámetros.
4. Agregar un Listener (Summary Report).
5. Ejecutar (botón “Start”).

📖 **Ejemplo básico:**

```
POST https://api.ejemplo.com/login
Body: {"username":"user1", "password":"123456"}
```

Automatización de pruebas de rendimiento en JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

📌 JMeter puede ejecutarse sin interfaz gráfica desde terminal, ideal para CI.

✅ **Comando básico CLI:**

- -n: modo no gráfico
- -t: archivo de prueba .jmx
- -l: log en formato .jtl
- -e -o: generación de reporte HTML

📂 **Esto genera un reporte navegable con métricas clave.**

```
jmeter -n -t tests/login-test.jmx -l resultados/resultados.jtl -e -o resultados/html
```

Configuración de pipeline en Jenkins para pruebas de rendimiento

aspasia
LA FORMACIÓN DE TU FUTURO

📌 JMeter puede ejecutarse en Jenkins mediante un Pipeline Script.

📖 **Ejemplo Jenkinsfile:**

```
pipeline {
  agent any
  stages {
    stage('Pruebas de Rendimiento') {
      steps {
        sh 'jmeter -n -t tests/login-test.jmx -l reportes/resultados.jtl'
        sh 'jmeter -g reportes/resultados.jtl -o reportes/html'
      }
    }
    stage('Publicar Reportes') {
      steps {
        publishHTML(target: [
          allowMissing: false,
          alwaysLinkToLastBuild: true,
          keepAll: true,
          reportDir: 'reportes/html',
          reportFiles: 'index.html',
          reportName: 'Reporte de JMeter'
        ])
      }
    }
  }
}
```

Integración de JMeter con herramientas de monitoreo

aspasia
LA FORMACIÓN DE TU FUTURO

📌 Puedes enviar métricas de JMeter a **InfluxDB** y visualizar en **Grafana**.

✅ **Componentes:**

- JMeter Backend Listener.
- InfluxDB como base de datos de tiempo real.
- Dashboard preconfigurado en Grafana.

Interpretación de reportes generados por JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

📌 **El reporte HTML de JMeter incluye:**

- Gráficas de tiempo de respuesta vs carga.
- Percentiles de respuesta (90%, 95%, 99%).
- Tasa de errores.
- Distribución de tiempos por transacción.

✅ **Elementos clave:**

- Response Time Over Time.
- Transactions Per Second.
- Response Codes.

📖 **Tips:**

- Percentil 95% < 800ms: excelente.
- Error Rate < 1%: aceptable.
- Throughput > 1000 req/min: ideal para APIs REST estándar.

Mejores prácticas para optimizar pruebas de rendimiento en JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

✅ **Recomendaciones clave:**

- **Separar entornos:** evitar pruebas en ambientes compartidos.
- **Datos variables:** usar CSV Data Set Config para simular usuarios reales.
- **Validaciones:** agregar Assertions para detectar respuestas erróneas.
- **Loops cortos para debug, largos para pruebas reales.**
- **Monitorear sistema:** CPU, RAM, disco y red durante las pruebas.

📖 **Caso común:**

En un ecommerce, se descubrió que a partir de 300 usuarios simultáneos, el login fallaba en un 12%. Se identificó un cuello de botella en la capa de autenticación y se optimizó antes del Black Friday.

aspasia
LA FORMACIÓN DE TU FUTURO

¿Cómo integrarías JMeter en un pipeline de CI/CD para garantizar los resultados de las pruebas?

aspasia
LA FORMACIÓN DE TU FUTURO

Ejercicio Guiado: Pruebas de Rendimiento con Apache JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

Ejercicio Guiado: Pruebas de Rendimiento con Apache JMeter

Las **pruebas de rendimiento** permiten evaluar cómo responde un sistema ante múltiples usuarios, solicitudes simultáneas o condiciones extremas. En este ejercicio usaremos **Apache JMeter** para simular múltiples peticiones a un servicio web público. Analizaremos métricas clave como **tiempo de respuesta, throughput y tasa de errores**.

aspasia
LA FORMACIÓN DE TU FUTURO

Objetivo

- Instalar y configurar Apache JMeter.
- Simular usuarios concurrentes sobre un endpoint REST.
- Capturar y analizar métricas de rendimiento.
- Generar reportes de resultados.
- (Opcional) Ejecutar las pruebas desde consola o CI.

🛠️ Paso 1: Instalación y Configuración de JMeter

aspasia
LA FORMACIÓN DE TU FUTURO

📌 ¿Qué haremos?
Instalaremos JMeter y lo configuraremos para comenzar nuestras pruebas.

Instrucciones:

1. Descarga JMeter desde:
https://jmeter.apache.org/download_jmeter.cgi

2. Descomprime el archivo y ejecuta JMeter:
   - En Windows: bin/jmeter.bat
   - En Linux/macOS: bin/jmeter.sh

3. Asegúrate de tener Java instalado (java -version en consola).

🧵 Paso 2: Crear un Plan de Prueba Básico

aspasia
LA FORMACIÓN DE TU FUTURO

📌 **¿Qué haremos?**
Diseñaremos una prueba de carga sobre una API REST que devuelve usuarios.

Instrucciones:

1. En el árbol de la izquierda haz clic derecho en Test Plan → Add > Threads (Users) > Thread Group
2. Configura el Thread Group con estos parámetros:
   - **Number of Threads (users):** 50
   - **Loop Count:** 5
   - **Ramp-Up Period:** 10 segundos

✅ Esto simula 50 usuarios enviando solicitudes, repitiendo el test 5 veces.

🌐 Paso 3: Agregar una Petición HTTP a la API

aspasia
LA FORMACIÓN DE TU FUTURO

📌 ¿Qué haremos?
Llamaremos un servicio REST para evaluar su capacidad de respuesta.

Instrucciones:

1. Clic derecho sobre el Thread Group →
   Add > Sampler > HTTP Request
2. **Configura:**
   - **Name:** API Usuarios
   - **Server Name or IP:** jsonplaceholder.typicode.com
   - **Method:** GET
   - **Path:** /users

✅ Este endpoint simula una API de usuarios como si fuera parte de una inmobiliaria.

📊 Paso 4: Agregar Listeners para ver resultados

aspasia
LA FORMACIÓN DE TU FUTURO

📌 ¿Qué haremos?
Usaremos componentes visuales para analizar el rendimiento de la prueba.

Instrucciones:

Haz clic derecho sobre el Thread Group y agrega los siguientes listeners:

- View Results in Table
- Summary Report
- Graph Results
- Aggregate Report

✅ Ejecuta la prueba con el botón ▶ **Start**.

📈 Paso 5: Interpretar los Resultados

aspasia
LA FORMACIÓN DE TU FUTURO

📌 **¿Qué haremos?**
Analizaremos las métricas más importantes del reporte.

Indicadores clave:

- **Average:** tiempo promedio de respuesta.
- **Min / Max:** tiempos extremos registrados.
- **Throughput:** solicitudes por segundo.
- **Error %:** porcentaje de errores.
- **Latency:** tiempo desde que se envía la solicitud hasta que se recibe respuesta.

💡 Puedes exportar los resultados como .csv para informes posteriores.

🧪 Paso 6: Ejecutar desde la línea de comandos (opcional)

aspasia
LA FORMACIÓN DE TU FUTURO

📌 ¿Qué haremos?
Ejecutaremos el test sin interfaz gráfica para automatizarlo en CI.

Instrucciones:

1. Guarda el test:
   File > Save Test Plan As > prueba-usuarios.jmx
2. Ejecuta en la terminal

```
jmeter -n -t /ruta/prueba-usuarios.jmx -l /ruta/resultados.jtl -e -o /ruta/html-report
```

Esto ejecuta la prueba y genera un reporte HTML visual.

✅ Puedes abrir el index.html del reporte generado.

aspasia
LA FORMACIÓN DE TU FUTURO

Resumen de lo aprendido

- **Tipos de Pruebas de Rendimiento:** Se estudiaron pruebas de carga, estrés y estabilidad para evaluar el comportamiento del sistema bajo.
- **Métricas y Herramientas:** Uso de métricas clave como latencia y throughput, junto a herramientas como JMeter para medir el rendimiento.
- **Implementación con JMeter:** Configuración de Thread Groups, ejecución de pruebas de carga y análisis de resultados mediante listeners e informes.
- **Automatización y Mejores Prácticas:** Integración de JMeter en pipelines con Jenkins, optimización de pruebas y uso conjunto con herramientas de monitoreo.

aspasia
LA FORMACIÓN DE TU FUTURO

Próxima clase...

Pruebas funcionales


dalid
aspasia
LA FORMACIÓN DE TU FUTURO