

Estrategia de Pruebas para el Microservicio #4

1. Objetivos de las Pruebas

- Asegurar que el microservicio de Notificaciones y Mensajería funcione correctamente y cumpla con los requisitos especificados.
- Identificar defectos y problemas antes de la implementación en producción.
- Validar la usabilidad, rendimiento, y seguridad del microservicio.

2. Enfoque de Pruebas

- **Pruebas Unitarias:** Utilizar herramientas como JUnit y Mockito para realizar pruebas sobre las funciones individuales, asegurando que cada componente cumpla con su objetivo.
- **Pruebas de Integración:** Implementar pruebas de integración con herramientas como Postman o Spring Test para verificar que los componentes interactúan correctamente entre sí.
- **Pruebas de Carga:** Usar herramientas como Apache JMeter o Gatling para simular altos volúmenes de tráfico y evaluar el rendimiento bajo condiciones de carga.
- **Pruebas de Seguridad:** Realizar pruebas de penetración y usar herramientas de análisis de seguridad como OWASP ZAP para identificar vulnerabilidades.
- **Pruebas de Usabilidad:** Realizar pruebas con usuarios reales para obtener feedback sobre la interfaz y la experiencia de usuario.
- **Pruebas de Regresión:** Implementar un conjunto de pruebas automatizadas que se ejecuten con cada nueva implementación para garantizar que las funcionalidades existentes no se vean afectadas.

3. Herramientas de Pruebas

- **Pruebas Unitarias:**
 - **JUnit:** Para realizar pruebas unitarias en el código Java.
 - **Mockito:** Para crear simulaciones y probar interacciones entre componentes.
- **Pruebas de Integración:**
 - **Postman:** Para pruebas de API y validación de respuestas.
 - **Spring Test:** Para pruebas de contexto de la aplicación.
- **Pruebas de Carga:**
 - **Apache JMeter:** Para simular carga en el sistema y analizar el rendimiento.
 - **Gatling:** Alternativa para pruebas de carga con un enfoque en simulaciones de usuarios.
- **Pruebas de Seguridad:**

- **OWASP ZAP:** Para realizar pruebas de seguridad y encontrar vulnerabilidades.
- **Pruebas de Usabilidad:**
 - **Sesiones de prueba con usuarios:** Reuniones con usuarios finales para evaluar la facilidad de uso del microservicio.

4. Cronograma de Pruebas

- **Fase de Planificación:** 1 semana
 - Definición de requisitos de prueba y preparación del entorno de prueba.
- **Desarrollo de Pruebas Unitarias:** 2 semanas
 - Implementación de pruebas unitarias y revisión del código.
- **Desarrollo de Pruebas de Integración:** 1 semana
 - Implementación de pruebas de integración y ajustes necesarios.
- **Ejecución de Pruebas de Carga:** 1 semana
 - Configuración y ejecución de pruebas de carga bajo diferentes escenarios.
- **Pruebas de Seguridad:** 1 semana
 - Realización de pruebas de seguridad y análisis de resultados.
- **Pruebas de Usabilidad:** 1 semana
 - Reuniones de feedback con usuarios y ajustes según sea necesario.
- **Pruebas de Regresión:** Continuas
 - Ejecución de pruebas de regresión en cada nueva implementación.

5. Recursos Necesarios

- **Equipo de Pruebas:** Personal con habilidades en pruebas de software, conocimiento de las herramientas mencionadas y capacidad para analizar resultados.
- **Infraestructura:** Entornos de prueba que simulen el entorno de producción, acceso a herramientas de pruebas y, si es necesario, dispositivos para pruebas de usabilidad.
- **Documentación:** Detallar todos los casos de prueba, resultados y defectos encontrados, asegurando una buena trazabilidad.

6. Criterios de Éxito

- Todos los casos de prueba críticos deben ser exitosos, con una tasa de fallos aceptable (por ejemplo, menos del 5%).
- Identificación y resolución de todos los defectos críticos antes de la implementación en producción.
- Retroalimentación positiva de los usuarios sobre la usabilidad del sistema.

7. Métricas de Pruebas

- **Tasa de éxito de pruebas unitarias:** Porcentaje de pruebas unitarias que pasaron.
- **Tasa de fallos de integración:** Porcentaje de pruebas de integración que pasaron.
- **Tiempo de respuesta bajo carga:** Medir el tiempo que tarda el sistema en responder bajo condiciones de alta carga.
- **Número de defectos encontrados:** Contar los defectos encontrados en cada fase de pruebas.