

# **Actividad 1. Evaluación de la calidad del software**

Plan de pruebas (versión mejorada con métricas y cobertura de compatibilidad)

**Estudiante:** Emanuel Bolaños Orozco

**Correo:** emabola01@gmail.com

**Teléfono:** +506 8497 6735

**Fecha:** 02/02/2026

## 1. Descripción del problema

Imagina que trabajas como tester en una empresa de desarrollo de software y te han asignado la tarea de evaluar un sistema de gestión de tareas usado por equipos de trabajo. El sistema permite crear, asignar, editar, eliminar y priorizar tareas, además de definir fechas límite. Incluye un panel de control con gráficos del progreso y rendimiento. Debe ser accesible desde navegadores web y dispositivos móviles, y se espera que sea rápido y fácil de usar.

## 2. Objetivo del plan de pruebas

Validar requisitos y el flujo principal del sistema (pruebas funcionales) y evaluar atributos de calidad (pruebas no funcionales).

## 3. Alcance

### Incluye:

- Gestión de tareas: crear, editar, eliminar, asignar, priorizar, fechas límite.
- Panel de control: actualización de gráficos ante cambios en tareas.
- Acceso web y vista responsive en móvil.
- Atributos: rendimiento (carga con más de 1000 tareas), usabilidad, compatibilidad y seguridad básica (autorización).

### No incluye:

- Integraciones externas (correo, calendarios, servicios de terceros) si existieran.
- Pruebas de recuperación ante desastres / alta disponibilidad (si no está en el alcance del curso).

## 4. Entorno de pruebas

El entorno debe acercarse a condiciones reales. A continuación se definen parámetros mínimos para que las pruebas de rendimiento y compatibilidad sean repetibles y comparables.

- **Ambiente:** Staging/QA con datos de prueba y configuración equivalente a producción (si aplica).
- **Datos:** base con ~1,200 tareas, ~50 usuarios, al menos 5 equipos/proyectos para simular carga real.
- **Navegadores (desktop):** Google Chrome v121-123, Mozilla Firefox v122-124, Microsoft Edge v121-123.
- **Móviles:** Android 13-14 (Chrome móvil), iOS 16-17 (Safari).
- **Dispositivos de referencia:** 360x640 (Android pequeño), 390x844 (iPhone), 412x915 (Android grande).
- **Equipo de prueba:** PC con 8 GB RAM mínimo; red estable (medir latencia si varía).

## 5. Criterios de entrada y salida

### Entrada:

- Requisitos disponibles y entendidos (historias de usuario o descripción funcional).
- Ambiente estable y accesible; datos de prueba cargados (más de 1000 tareas).
- Casos de prueba definidos y priorizados.

### Salida:

- Se ejecutaron todos los casos de prueba de prioridad Alta.
- Defectos críticos y altos corregidos o con plan de mitigación aceptado.
- Resultados de rendimiento contra métricas definidas o desviaciones documentadas.

## 6. Pruebas funcionales (casos principales)

ID	Módulo	Caso de prueba	Prioridad
F01	Tareas	Crear una tarea	Alta
F02	Tareas	Eliminar una tarea existente	Alta
F03	Tareas	Editar una tarea existente	Alta
F04	Asignación	Asignar una tarea a un miembro del equipo	Alta
F05	Prioridad	Establecer prioridad a cada tarea	Media
F06	Panel de gráficos	Las gráficas reflejan cambios al crear/completar tareas	Media

## 7. Pruebas no funcionales (criterios de aceptación)

ID	Módulo	Criterio de aceptación
NF01	Rendimiento	Con 1,000+ tareas, la carga inicial y navegación mantienen tiempos de respuesta dentro de métricas definidas (ver sección 8).
NF02	Compatibilidad	Funciona en Chrome v121-123, Firefox v122-124 y Edge v121-123; vista responsive y funciones principales en Android 13-14 (Chrome) e iOS 16-17 (Safari).
NF03	Usabilidad	Usuarios nuevos completan el flujo básico (crear y asignar tarea) sin bloqueos; interfaz clara y consistente.
NF04	Seguridad	Autorización y control de acceso: un usuario no puede ver/editar tareas de equipos ajenos; sesiones protegidas.

## 8. Métricas de rendimiento (carga y respuesta)

Para evaluar NF01 de forma objetiva, se definen métricas medibles y umbrales.

- **Métrica principal:** tiempo de respuesta (p95) por acción crítica.
- **Secundarias:** throughput (req/s), tasa de error (%), uso de CPU/RAM del servidor, y tiempos de carga de interfaz.
- **Medición recomendada:** 3 ejecuciones por escenario y promediar resultados; registrar p50 y p95.

Escenario	Carga	Métrica objetivo (umbral)
Carga inicial (lista de tareas)	Dataset 1,200 tareas; 50 usuarios	p95 <= 3.0 s; errores < 1%
Navegación + filtros/búsqueda	100 usuarios concurrentes	p95 <= 2.0 s por acción; errores < 1%
Crear/editar/asignar tarea	200 usuarios concurrentes	p95 <= 1.5 s por acción; errores < 1%

**Detalle del entorno para rendimiento:** anotar hardware del servidor (CPU/RAM), base de datos (tipo/versión), y red. Esto ayuda a repetir la prueba y comparar resultados.

## 9. Cobertura de compatibilidad (matriz sugerida)

Para completar NF02 con más precisión, se documentan versiones y dispositivos de prueba.

Plataforma	Navegador / SO	Versiones	Pruebas mínimas
Desktop	Chrome	v121-123	F01-F04, F06
Desktop	Firefox	v122-124	F01-F04, F06
Desktop	Edge	v121-123	F01-F04, F06
Móvil	Android (Chrome)	13-14	F01-F04 + responsive
Móvil	iOS (Safari)	16-17	F01-F04 + responsive

## 10. Pruebas manuales vs. automatizadas

La selección busca balance: automatizar lo repetitivo/crítico y mantener manual lo visual/experiencial.

Área	Manual	Automatizada	Justificación
Tareas (crear/editar/eliminar)		X	Flujo crítico y repetitivo: ideal para regresión (UI o API).
Asignación y permisos		X	Alta prioridad y sensible a cambios: automatizar reduce errores.
Panel de gráficos	X		Validación visual (coherencia de gráficos) es más confiable manualmente.
Usabilidad	X		Requiere observación y feedback real del usuario.
Rendimiento		X	Métricas repetibles y ejecuciones controladas (k6/JMeter).
Compatibilidad	X	Opcional	Manual en navegadores clave; se puede complementar con BrowserStack/Sauce Labs.
Seguridad básica	X	X	Manual para roles; automatizable con escáneres (OWASP ZAP) desde etapas tempranas.

## 11. Herramientas recomendadas (opcional)

- **Automatización UI:** Cypress o Playwright.
- **Automatización API:** Postman/Newman o pruebas con scripts.
- **Rendimiento:** k6 o JMeter (carga/concurrencia, métricas p95).
- **Web performance:** Lighthouse (LCP, TTI) si aplica a frontend.
- **Seguridad temprana:** OWASP ZAP (escaneo básico automatizable).
- **CI/CD:** GitHub Actions/Jenkins para ejecutar regresión automática.

## 12. Plantilla de reporte de defectos (mini)

Campo	Ejemplo
ID	BUG-001
Título	No permite asignar tarea a usuario del equipo
Severidad / Prioridad	Alta / Alta
Pasos para reproducir	1) Ingresar 2) Crear tarea 3) Asignar 4) Guardar
Resultado esperado	La tarea queda asignada y visible en el tablero
Resultado real	Muestra error 500 y no guarda
Evidencia	Captura o video; logs si aplica