Reporte de Pruebas Web

| **Información del proyecto** | |
| --- | --- |
| Nombre del proyecto: TaskManager | |
| Versión del documento: 1.0 | Fecha: 08/12/2024 |
| Preparado por:   * [Isael Alejandro Reyes Vargas](mailto:20233tn160@utez.edu.mx) * [Danna Paola Sanchez Martinez](mailto:20213tn039@utez.edu.mx) * [Miguel Angel Sanchez Perez](mailto:20213tn040@utez.edu.mx) * [Angel Gabriel Aguilar Garcia](mailto:20233tn188@utez.edu.mx) | |
| **Resumen ejecutivo** | |
| El presente reporte de pruebas tiene como objetivo documentar los resultados obtenidos durante el proceso de verificación y validación de la aplicación móvil TaskManager, desarrollada como una práctica académica. Esta aplicación permite a los usuarios subir y visualizar tareas de una universidad, almacenándolas en una base de datos Firebase en tiempo real. El propósito es ofrecer evidencia del correcto funcionamiento de las funcionalidades clave, así como identificar cualquier área de mejora. | |
| **Alcance de pruebas** | |
| **Tipos de pruebas:**   * Funcionales: Validación del correcto funcionamiento de la creación y visualización de tareas. * Usabilidad: Evaluación de la experiencia del usuario, facilidad de uso y navegación intuitiva. * Rendimiento: Medición de tiempos de respuesta (objetivo: <2s) al añadir tareas. * Seguridad: Verificación de la comunicación con Firebase, cifrado de datos y acceso no autorizado. * Consumo de recursos: Monitoreo del uso de CPU, memoria y batería durante la adición de tareas. * Accesibilidad: Comprobación con lectores de pantalla y navegación mediante gestos para usuarios con discapacidad visual. | |
| **Funcionalidades a prueba:**   * Añadir tareas (Título y Descripción) * Visualizar las tareas guardadas en Firebase * Mensajes de confirmación (e.g. "Tarea agregada") | |
| **Entornos utilizados:**   * Plataforma: Android (versión 11 y 12) * Dispositivo principal: Pixel 8 Pro * Conexión a Internet estable (>30mbps) * Base de datos: Firebase (en entorno de prueba) * Lenguaje: Kotlin/Java (dependiendo de la implementación base) | |
| **Metodología** | |
| **Herramientas utilizadas:**   * Appium: Para pruebas funcionales automatizadas. * Fing: Para monitoreo de red. * Profiler (Android Studio): Para consumo de recursos. * APK Analyzer: Revisión de permisos. * Wireshark/Fiddler: Monitoreo de tráfico y cifrado. * JIRA: Herramienta de seguimiento de incidencias. | |
| **Pruebas unitarias:**  JUnit  Simulación de un entorno real: Mockito | |
| **Pruebas funcionales:**  Ejecución automatizada de casos de prueba con Cypress.  Validación del correcto funcionamiento de las funcionalidades clave del sistema. | |
| **Pruebas de usabilidad:**  Evaluación con un grupo de usuarios, observación de interacción, testing en diferentes navegadores para Checklist | |
| **Pruebas de seguridad:**  Escaneo de vulnerabilidades con OWASP ZAP, revisión de contraseñas seguras, inyecciones SQL, XSS, así como pruebas automatizadas y por módulos del backend usando Cypress y JUnit. | |
| **Pruebas de rendimiento:**  Cargas simuladas con JMeter para medir tiempos de respuesta (<2s) y estabilidad del sistema bajo estrés. | |
| **Pruebas unitarias:**  Ejecución con JUnit para validar lógica interna de módulos y funciones. | |
| **Criterios de entrada:**   * Código fuente completo y revisado. * Entorno de pruebas operativo. * Casos de prueba documentados, revisados y aprobados. | |
| **Criterios de salida:**   * Todas las pruebas funcionales ejecutadas sin defectos críticos. * Cobertura de código de al menos el 80%. * Rendimiento aceptable (menor a 2s) y sin defectos críticos pendientes. * Satisfacción del usuario en usabilidad mayor al 80%. | |
| **Resultados detallados** | |
| * Con un total de 41 casos de prueba, se cubrieron las áreas funcionales clave del sistema. * Los resultados preliminares indican que el sistema cumple con la mayoría de las funcionalidades, sin defectos críticos pendientes. El tiempo de respuesta promedio se mantuvo por debajo de los 2 segundos en las pruebas de carga iniciales. La evaluación de usabilidad mostró una satisfacción superior al 80%. | |
| **Tasas de estado desglose por tipo de prueba:**   * **Pruebas Funcionales:**   + 100% ejecutadas, 100% aprobadas.   + Total de casos de prueba realizados: 13   + Casos de éxito: 13   + Casos de fallo: 13 * **Pruebas de Usabilidad:**   + 100% ejecutadas, 60% aprobadas.   + Total de casos de prueba realizados: 10   + Casos de éxito: 6   + Casos de fallo: 4 * **Pruebas de Rendimiento:**   + 100% ejecutadas, tiempos menores a 2 segundos .   + Total de casos de prueba realizados: 3   + Casos de éxito: 3   + Casos de fallo: 3 * **Pruebas de Seguridad:**   + 100% ejecutadas, sin vulnerabilidades críticas.   + Total de casos de prueba realizados: 6   + Casos de éxito: 6   + Casos de fallo: 6 * **Pruebas Unitarias:**   + Cobertura 90% del sistema, 100% aprobadas.   + Total de casos de prueba realizados: 8   + Casos de éxito: 8   + Casos de fallo: 8   **Tasa de estado general:**   * **Tasa de éxito:**   + 90.24% (37 de 41 casos pasaron) * **Tasa de fallo:**   + 9.75% (4 de 41 casos fallaron)   **Detalles de pruebas:**  [Informe Final de pruebas](https://docs.google.com/spreadsheets/d/1QWlo9OAIcha-s5Halfaa3JYONoCL2y7EosRmgdH-63c/edit?usp=sharing)  **Detalles de fallos: Casos de usabilidad**   1. Validaciones visuales incompletas en formularios:    * Al momento de ingresar datos, algunos campos del formulario presentan un borde de color rojo, indicando una condición de error o falta de información. Sin embargo, esta retroalimentación visual no se implementa de forma consistente en todos los formularios disponibles en el sistema. Como resultado, el usuario puede enfrentarse a situaciones en las que no reciba ninguna señal visual clara de que un dato es requerido o no cumple con las condiciones esperadas. Esta falta de uniformidad dificulta la experiencia del usuario y reduce la claridad sobre los pasos necesarios para completar correctamente el registro. 2. Menú con etiquetas sin íconos representativos:    * La navegación principal del sistema está compuesta únicamente por etiquetas textuales que conducen a las diferentes funcionalidades. Si bien, al hacer clic en cada opción, se accede de manera apropiada al módulo correspondiente, la ausencia de íconos reconocibles limita la capacidad del usuario de identificar rápidamente la sección que desea consultar. Esto puede aumentar el tiempo de familiarización con la interfaz y reducir la eficiencia en la navegación, especialmente para usuarios nuevos o poco experimentados. 3. Falta de atajos de teclado:    * El sistema no ofrece atajos de teclado para agilizar la interacción con la interfaz. Esto significa que los usuarios, especialmente aquellos con mayor conocimiento o que trabajan regularmente con la aplicación, no cuentan con una manera rápida de acceder a funciones críticas, cambiar de módulos o completar tareas sin depender exclusivamente del ratón. La ausencia de atajos limita la productividad y la accesibilidad, impactando tanto a usuarios avanzados como a aquellos con necesidades especiales. 4. Ausencia de una sección de ayuda o preguntas frecuentes:    * El sistema no incluye una sección de ayuda, soporte o preguntas frecuentes que oriente al usuario en caso de dudas, dificultades o desconocimiento sobre el funcionamiento de las funcionalidades. Esto implica que, ante cualquier problema o incertidumbre, el usuario debe buscar orientación externa o recurrir a ensayo y error, lo que puede generar frustración, ralentizar el aprendizaje y disminuir la satisfacción general con el producto. Una sección de ayuda integrada brindaría claridad, reduciría la carga del servicio de soporte y mejoraría significativamente la experiencia del usuario final. | |

| **Cronograma** |
| --- |
| * Inicio pruebas funcionales: 05/12/2024 * Inicio pruebas rendimiento: 08/12/2024 * Inicio pruebas consumo recursos: 09/12/2024 * Inicio pruebas usabilidad: 08/12/2024 * Inicio pruebas seguridad: 08/12/2024 * Inicio pruebas unitarias: 08/12/2024 * **Finalización: 10/12/2024** |
| **Casos de prueba** |
| [Casos de prueba](https://docs.google.com/document/d/1RyCmcjDT8YO8KvicZxdHsK2agxU-lJOB/edit?usp=sharing&ouid=108998223734136559569&rtpof=true&sd=true) |
| **Análisis de defectos** |
| **Severidad Alta**   * Verificar que el sistema evita errores al completar un formulario de registro con campos vacíos.   Área: Seguridad  Impacto: Permite realizar registros sin una validación previa, compromete al sistema a ataques XSS.  **Severidad Media**   * Verificar que el sistema utiliza íconos y etiquetas claras en el menú de navegación para que los usuarios puedan identificar fácilmente las funcionalidades disponibles.   Área: Usabilidad  Impacto: Compromete parcialmente la experiencia del usuario ya que podría ser difícil navegar por el sistema.   * Verificar que el sistema permite realizar acciones comunes usando atajos de teclado.   Área: Usabilidad  Impacto: Puede generar retrasos o una sensación de navegación más lenta al usar el sistema.   * Verificar que cada módulo incluye un botón de ayuda con tutoriales específicos.   Área: Usabilidad  Impacto: Confusión por parte del usuario al no contar con una sección de ayuda o preguntas frecuentes para el usuario final. |

**Riesgos y mitigaciones**

| **Riesgo** | **Impacto** | **Probabilidad** | **Estado** | **Estrategia de mitigación** |
| --- | --- | --- | --- | --- |
| Cambios en los requerimientos funcionales | Bajo | Medio | Resuelto | Reuniones constantes con el cliente que solicitó el proyecto. |
| Contratiempos en el desarrollo del sistema | Alto | Medio | Resuelto | Tener una gestión del tiempo más optimizada para evitar problemas de retrasos. |

| **Gestión de defectos** |
| --- |
| **Herramientas de seguimiento:**  **JIRA:** Software para seguimiento de errores e incidencias  **Flujo de trabajo:**   1. Reportar por parte del equipo de QA. 2. Asignar a un equipo de desarrollo especializado. 3. Resolución del defecto por parte del equipo. 4. Verificación por parte del equipo de QA para confirmar las correcciones. 5. Cerrar si el defecto fue solucionado exitosamente.   **Priorización:**  Crítico > Alto > Medio > Bajo |
| **Métricas de pruebas** |
| **Cobertura de pruebas:**  Cobertura total: 83% del sistema  Objetivo estimado: 95% del sistema  Por cuestiones de tiempo y fechas de entrega del proyecto, las pruebas aunque lograron una alto grado de cobertura, no pudieron concluir con el porcentaje esperado.  **Eficiencia de las pruebas:**  Se encontraron un total de 4 errores en 5 días lo que nos da un promedio de 0.8 errores por día.  Objetivo estimado: 5 errores por día  En relación a un bajo promedio de defectos en las áreas analizadas, el promedio de fallas encontradas en el sistema se vio reducido drásticamente.  **Densidad de defectos:**  Baja 0.02 (menos de 1 defecto crítico por módulo crítico)  Objetivo estimado: 0.025  Se obtuvo un valor de fallos críticos menor a la esperada.  **Tasa de re-apertura de defectos:**  Menor al 5%  Objetivo estimado: 10%  En consecuencia a la baja tasa de defectos encontrados durante la realización de las pruebas y a la rápida resolución por parte de los desarrolladores, se obtuvo un bajo nivel de re apertura de defectos. |
| **Conclusión y recomendaciones** |
| El sistema hospitalario cumple con los criterios funcionales, de rendimiento y de forma parcial de usabilidad establecidos. No se han detectado vulnerabilidades de seguridad críticas. Se recomienda cerrar el ciclo de pruebas, con un mantenimiento continuo y reevaluaciones periódicas de rendimiento y seguridad para garantizar la estabilidad a largo plazo. |

| **Glosario** | | | | |
| --- | --- | --- | --- | --- |
| Apache Tomcat | Servidor web y contenedor de servlets en Java que permite ejecutar aplicaciones web desarrolladas con tecnología Java. | | | |
| Base de datos (MySQL) | Sistema de gestión de bases de datos relacional, utilizado para almacenar y gestionar información de forma estructurada. | | | |
| Bootstrap | Framework de CSS, HTML y JavaScript que facilita el diseño de interfaces web responsivas y atractivas sin tener que escribir todo el código desde cero. | | | |
| Cypress | Herramienta de automatización de pruebas para aplicaciones web, que permite ejecutar casos de prueba funcionales y de integración en el navegador. | | | |
| Spring Web | Soporte para el desarrollo de aplicaciones web y REST. | | | |
| Spring Data JPA | Facilita el acceso y manipulación de datos en bases de datos relacionales. | | | |
| Spring Security | Marco de trabajo para gestionar la seguridad y control de acceso en aplicaciones Java. | | | |
| JIRA | Herramienta de seguimiento de incidencias y administración de proyectos, utilizada para reportar, asignar y gestionar defectos en el software. | | | |
| Apache JMeter | Herramienta de pruebas de rendimiento y carga que simula múltiples usuarios accediendo a una aplicación para medir su respuesta y estabilidad bajo condiciones de estrés. | | | |
| JUnit | Framework de pruebas unitarias para Java, que permite verificar el correcto funcionamiento de pequeñas partes del código (métodos o funciones). | | | |
| Mockito | Herramienta para crear objetos simulados (mocks) y así probar el comportamiento de componentes individuales sin depender de otros módulos del sistema. | | | |
| OWASP ZAP | Herramienta de seguridad para analizar aplicaciones web, detectar vulnerabilidades y revisar la protección contra ataques comunes. | | | |
| XSS | Tipo de vulnerabilidad de seguridad en aplicaciones web que permite inyectar código malicioso en páginas vistas por otros usuarios. | | | |