

Propuesta para implementar artefactos en la fase de pruebas del módulo registro y autenticación de usuarios en el desarrollo de sistemas online

¹Mtro. Ivan Rafael Sánchez Juárez, ²Mtra. Maria Petra Paredes Xochihua, ³Dr. Portilla, Flores-Alberto

1 y 2 Tecnológico Nacional de México / ITS de San Martín Texmelucan

3 Responsable Área de Innovación e Investigación Software, Softura Solutions

Resumen - Este artículo presenta una propuesta para fortalecer la gestión de pruebas en el desarrollo de sistemas online, enfocándose en la integración de planes de pruebas y casos de prueba como artefactos esenciales específicamente para el módulo de registro y autenticación de usuarios. La investigación se centra en la importancia de una estrategia de pruebas sólida y bien documentada en el desarrollo de los módulos mencionados.

Además, se destaca la incorporación de casos de prueba como elementos esenciales en esta metodología, lo que permite una mayor coherencia y seguimiento en la fase de pruebas. La propuesta subraya cómo estos elementos, cuando se utilizan de manera conjunta y sistemática, mejoran la calidad del software, reducen errores y fomentan una comunicación efectiva entre los equipos de desarrollo y pruebas. Este enfoque se presenta como una contribución significativa para optimizar la gestión de pruebas en el desarrollo de sistemas, mejorando la eficiencia y la calidad del software entregado.

Palabras clave - artefacto, calidad, prueba, registro de usuarios, software

Abstract: This article presents a proposal to strengthen test management in the development of online systems, focusing on the integration of test plans and test cases as essential artifacts specifically for the user registration and authentication module. The research focuses on the importance of a solid and well-documented testing strategy in the development of the mentioned modules.

Furthermore, the incorporation of test cases as essential elements in this methodology stands out, which allows greater coherence and monitoring in the testing phase. The proposal highlights how these elements, when used together and systematically, improve software quality, reduce errors, and foster effective communication between development and testing teams. This approach is presented as a significant contribution to optimizing test management in systems development, improving the efficiency and quality of the delivered software.

Keywords: artifact, quality, software, test, user registration

I. INTRODUCCION

El desarrollo de sistemas de software se divide en las siguientes fases: análisis de requerimientos, diseño, desarrollo, pruebas e implementación. La fase de pruebas desempeña un papel fundamental en la búsqueda de la calidad y eficiencia del software, la importancia de la fase radica en la detección de errores, mejora de la calidad, ahorro de tiempo y recursos, seguridad y por último, en el correcto cumplimiento de los requerimientos. En esta fase se debe destinar al menos un 25% del tiempo dedicado al proyecto para determinar el éxito del mismo, el departamento de pruebas debe realizar algunas de las siguientes operaciones: plan de pruebas, diseño de casos de prueba, ejecución de casos de prueba, así como la validación de las pruebas.

Los errores más comunes en QA (Quality Assurance) Testing considerados en [2] son: “Uno de los errores más comunes en QA Testing es la falta de una planificación adecuada y una documentación clara. Sin

una estrategia de prueba bien definida y un plan de pruebas completo, los equipos de QA pueden perderse en su enfoque y no cubrir todos los aspectos necesarios. La falta de documentación también puede dificultar la colaboración con otros miembros del equipo y la comprensión de los resultados de las pruebas.”

En este artículo se presenta la propuesta de artefactos, como herramientas clave que pueden revolucionar la forma en que las organizaciones abordan las pruebas de software. Analizando su importancia y los beneficios que aporta al proceso de desarrollo de sistemas, la necesidad de documentación rigurosa y la integración de artefactos como elementos esenciales para mejorar la eficiencia, reducir errores y fortalecer la comunicación entre equipos de desarrollo y pruebas.

II. METODOLOGÍA

La metodología a seguir para la elaboración de la propuesta consiste en los siguientes pasos: planificación de pruebas, diseño de casos de prueba, preparación del entorno de pruebas, ejecución de pruebas, reporte y gestión de proyectos. En este artículo se presenta la planificación de pruebas y el diseño de casos de prueba. [4] indica que la falta de un buen plan de pruebas y la desconexión entre los requerimientos del negocio y lo que se prueba, es una de las principales barreras a la hora de entregar software que cumpla con los objetivos de negocio.

Antes de llevar a cabo el plan de pruebas es importante tener en consideración cuál será el alcance, comprender de manera clara, cuáles son los requerimientos y especificaciones que tendrá el sistema, designar a un equipo dedicado a ejecutar las pruebas y contar con un calendario de trabajo.

Adicionalmente, [1] menciona lo siguiente: Para que su proyecto sea un éxito, debe generar y colocar criterios de prueba en el plan de pruebas. Un buen conjunto de pautas debe incluir lo que necesita probarse, así como diferentes niveles de calidad que se considerarán aceptables.

En el diseño de casos de prueba consiste en crear una serie de casos detallados que describen cómo se probará una función o un punto específico del software. Los casos de prueba ayudan a garantizar que el software funcione correctamente y cumpla con los requisitos establecidos.

Algunas de las actividades que se desarrollan en esta etapa son: Identificación de requisitos de prueba, pasos de prueba, datos de entrada, resultados esperados,

criterios de aceptación o rechazo, resultados obtenidos, seguimiento de errores.

La forma de escribir pruebas y casos de prueba eficaces se puede optimizar con el tiempo. Algunos y las mejores prácticas incluya el uso de títulos sólidos, descripciones sólidas y mantener el lenguaje conciso y claro. Pero también querrá incluir condiciones previas, suposiciones y los resultados esperados. Toda esta información es relevante para el probador de software, especialmente cuando se determina si el caso de prueba debe ser un "aprobado" o un "error" en su lugar. [3].

III. RESULTADOS Y DISCUSIÓN

El primer punto desarrollado es el plan de pruebas, se define el objetivo del plan de pruebas, el alcance, requerimientos funcionales y no funcionales, recursos, calendario, tipos de pruebas (funcionalidad, seguridad, rendimiento, usabilidad y compatibilidad), criterios de aceptación y finalmente, el informe de resultados. En la figura 1 se muestra el objetivo, alcance y requerimientos.

PLAN DE PRUEBAS	
Objetivo: Asegurar que el sistema de servicios en línea cumple con los requisitos funcionales y no funcionales, es seguro y proporciona una experiencia de usuario óptima	
Alcance: El plan de pruebas abarcará todas las funcionalidades clave del sistema de servicios en línea, desde el acceso y utilización de servicios hasta la gestión de cuentas de usuario.	
Requisitos del sistema:	
Requisitos funcionales:	Requisitos no funcionales:
<ul style="list-style-type: none"> • Registro de usuarios • Inicio de sesión • Selección y acceso a productos o servicios • Reserva de servicios • Pago por servicios • Calificación y reseñas de servicios • Recuperación de contraseña • Gestión de cuentas de usuario 	<ul style="list-style-type: none"> • Rendimiento: debe ser capaz de manejar un mínimo de 500 transacciones por minuto con un tiempo de respuesta promedio no superior a 2 segundos. • Seguridad: Garantizar que los datos de los usuarios y las transacciones sean seguros. • Compatibilidad: Asegurarse de que el sistema sea compatible con varios navegadores y dispositivos.

Figura 1: Plan de pruebas

En la figura 2 se muestran los recursos necesarios para ejecutar las pruebas, también, se muestra la calendarización por semanas de la fase de pruebas.

Recursos:	Calendario:
<ul style="list-style-type: none"> • Equipo de pruebas • Datos de prueba • Entorno de pruebas • Herramientas de automatización de pruebas 	Semana 1: Planificación: Semana 2: Preparación de entorno de pruebas y datos Semanas 3-6: Ejecución de pruebas: Semana 7: Informe de resultados: Semana 8-9: Corrección de errores y pruebas de regresión:

Figura 2: Recursos y calendarización.

Los tipos de pruebas realizados son de funcionalidad, seguridad, rendimiento, usabilidad y compatibilidad. Dentro de cada tipo de prueba, se indican los puntos a probar, como puede ser pruebas de registro de usuarios (funcionalidad), autenticación (seguridad), estrés (rendimiento) y facilidad de acceso (usabilidad). En la figura 3 se muestran los tipos y elementos de pruebas.

TIPOS DE PRUEBAS:	
Pruebas de funcionalidad: <ul style="list-style-type: none"> • Pruebas de registro de usuarios • Pruebas de inicio de sesión • Pruebas de selección y acceso a productos o servicios • Pruebas de reserva de servicios • Pruebas de proceso de pago 	Pruebas de rendimiento: <ul style="list-style-type: none"> • Pruebas de carga • Pruebas de estrés
Pruebas de seguridad: <ul style="list-style-type: none"> • Pruebas de seguridad de datos del usuario • Pruebas de seguridad de transacciones • Pruebas de seguridad de autenticación 	Pruebas de usabilidad: <ul style="list-style-type: none"> • Pruebas de facilidad de uso • Pruebas de accesibilidad
	Pruebas de compatibilidad: <ul style="list-style-type: none"> • Pruebas de compatibilidad con navegadores • Pruebas de compatibilidad con dispositivos móviles

Figura 3: Tipos de pruebas

Finalmente, se colocan los criterios de aceptación de acuerdo a los diferentes tipos de seguridad, así como el informe de resultados obtenidos en la fase de pruebas. En la figura 4 se presenta los últimos datos del plan de pruebas.

Criterios de aceptación:
<ul style="list-style-type: none"> • Todos los casos de prueba deben pasar sin errores críticos. • El rendimiento del sistema debe ser aceptable bajo condiciones de carga máxima. • Los datos de los usuarios y las transacciones deben estar seguros en todo momento. • La interfaz de usuario debe ser intuitiva y fácil de usar.
Informe de resultados:
<ul style="list-style-type: none"> • Resumen de resultados de pruebas • Errores identificados con detalles • Recomendaciones para corrección • Lista de pruebas completadas

Figura 4: criterios de aceptación e informe de resultados

Para la realización de pruebas, se implementaron los artefactos plan de pruebas unitarias y la matriz de trazabilidad. En la tabla 1 se muestra la hoja de control generada para el plan de pruebas unitarias. La hoja de control contiene los siguientes datos: Nombre de la empresa, nombre del documento, Proyecto, Módulo, Sprint, Entregable, Autor, Versión/Edición, Aprobado por, Fecha Versión, Fecha Aprobación, No. Total de Páginas y la sección de registro de cambios (versión, causa de cambio, responsable de cambio y fecha del cambio)

<<Empresa>>			
Plan de Pruebas Unitarias			
Proyecto	<<Nombre Proyecto>>		
Módulo	<<Nombre del módulo>>		
Sprint	<<Número del Sprint>>		
Entregable	Plan de Pruebas Unitarias		
Autor	Equipo de pruebas	Fecha Versión	DD/MM/AAAA
Versión / Edición	1.0	Fecha Aprobación	DD/MM/AAAA
Aprobado Por		N° Total de Páginas	X

REGISTRO DE CAMBIOS

Versión	Causa del cambio	Responsable del cambio	Fecha del cambio
1.0	Versión Inicial	Apellido paterno- Apellido materno, Nombre(s)	DD/MM/AAAA

Tabla 1: Hoja de control

Posteriormente se generó el plan de pruebas unitarias con los siguientes datos: Nombre de la empresa, Nombre del documento, Número del caso de prueba, sección, prerequisites y la sección para describir los casos de prueba (No. Descripción, Datos de entrada, Salida esperada, Aprobado – Si, NO – y Observaciones). En la tabla 2 se observa el contenido de este artefacto.

<<Empresa>>

Plan de Pruebas Unitarias

Número del Caso de Prueba	Módulo	Prerrequisitos
CP001	Perfil de un negocio	Haber registrado el negocio e iniciado sesión

CP001					
N°	Descripción	Datos de entrada	Salida esperada	Aprobado	Observaciones
1	Botón para regresar a la página anterior	Dar clic sobre el botón	Mostrar la página principal del negocio	No	Redirecciona a la página principal de Google
2	Cambiar la imagen del logotipo del negocio	Dar clic en el botón de examinar para buscar una imagen con formato jpg y tamaño de 500x500 pixeles	Agregar a la página de perfil, el nuevo logotipo de la empresa	Si	Agrega de manera adecuada la imagen del logotipo
3					
4					

Tabla 2: Plan de Pruebas Unitarias

Finalmente, en el artefacto de Matriz de trazabilidad generado, se tiene la siguiente información: Nombre del documento, Nombre del proyecto, Fecha de revisión del documento, Elaboró y la sección de trazabilidad con los siguientes datos:

- # Caso de prueba
- Descripción.
- Datos de entrada.
- Criterios de aceptación.
- Requisitos de Ambiente/sistema
- Usuario que ejecuta la prueba – ciclo 1
- Fecha de ejecución Prueba – Ciclo 1
- Resultado obtenido – Ciclo 1
- Estado de la prueba – ciclo 1 (Aprobado o Rechazado)
- Usuario que ejecuta la prueba – ciclo 2
- Fecha de ejecución prueba – ciclo 2
- Resultado obtenido – ciclo 2
- Estado de la prueba – ciclo 2 (Aceptado o Rechazado)
- Observaciones

En la tabla 3 se muestra la matriz de trazabilidad del requerimiento: “El usuario deberá ingresar con su correo electrónico y contraseña, en caso de que se le olviden sus datos de acceso deberá mandarle a una pantalla para poder recuperarlos” y las columnas: #Caso de prueba, descripción del caso de prueba, datos de entrada, criterios de aceptación, requisitos de ambiente/sistema y el Usuario que ejecuta la prueba en el Ciclo 1.

# Caso de Prueba	Descripción	Datos de entrada	Criterios de aceptación	Requisitos de Ambiente/sistema	Usuario que ejecuta la prueba
CP001	Ingresar a la pantalla de inicio de sesión, escribir correo electrónico y contraseña. El correo electrónico debe estar validado y no se ingresó la contraseña	Correo electrónico válido. Clic en el botón Ingresar	El sistema debe mandar un mensaje solicitando el campo de contraseña		Sánchez-Juárez, Ivan Rafael

Tabla 3: Matriz de Trazabilidad (primera parte)

La tabla 4 muestra la continuación de la matriz de trazabilidad, ahora con las columnas: fecha ejecución prueba – ciclo 1, Resultado obtenido – ciclo 1, Estado de la prueba ciclo 1.

Usuario que ejecuta la prueba - Ciclo 1	Fecha Ejecución Prueba - Ciclo 1	Resultado Obtenido - Ciclo 1	Estado de la prueba - Ciclo 1
Sánchez-Juárez, Ivan Rafael Tester	20/07/2023	El sistema no validó el campo	Rechazado

Tabla 4: Matriz de Trazabilidad (segunda parte)

En caso de que la prueba haya sido rechazada, se llenarán las siguientes columnas de la matriz: Usuario que ejecuta la prueba ciclo 2, fecha ejecución prueba ciclo 2, resultado obtenido ciclo 2, estado de la prueba ciclo 2 y finalmente, observaciones. La tabla 5 muestra el contenido de éste caso de prueba.

Usuario que ejecuta la prueba - Ciclo 2	Fecha Ejecución Prueba - Ciclo 2	Resultado Obtenido - Ciclo 2	Estado de la prueba - Ciclo 2	Observaciones
Maria Petra Paredes Xochihua	21/07/2023	El sistema validó el campo y mostró el mensaje para solicitar la contraseña	Aprobado	Con ésta última prueba, queda liberado el requerimiento funcional RF001

CONCLUSIÓN

La fase de pruebas en el desarrollo de software es una etapa muy importante para poder entregar un producto de calidad, entre las ventajas que se tiene es la identificación de errores y defectos antes de salir a producción.

Con los artefactos desarrollados ayudará a las fábricas de software a prevenir fallos en un sistema cuando ya se encuentre en el despliegue de producción, sin embargo, se debe hacer un artefacto por cada módulo para detectar los errores, cada módulo cuenta con sus propios casos de prueba, datos de entrada y validaciones, dependiendo del tipo de sistema a implementar, por lo que se recomienda utilizar las plantillas y adecuarlas a las necesidades de cada proyecto y módulo en desarrollo.

RECONOCIMIENTO

Agradecemos al Tecnológico Nacional de México / ITS de San Martín Texmelucan por todo el apoyo brindado para la realización de este proyecto, de igual manera se agradece a la empresa Softura Solutions por permitirnos realizar las estadías técnicas en sus instalaciones.

REFERENCIAS

- [1] Andrade, Alex. (10 de septiembre de 2021). Cómo desarrollar un plan de pruebas sólido pero simple. <https://alexandrade.net/blog-de-ingenieria-de-software/calidad-de-software/como-desarrollar-un-plan-de-pruebas-solido-pero-simple/>
- [2] BEDU. (20 de abril de 2023). Errores comunes en QA Testing y cómo evitarlos. <https://bedu.org/blog/tecnologia/errores-comunes-en-qa-testing-y-como-evitarlos>
- [3] Mc Mullin, William. (27 de mayo de 2021). Cómo escribir casos de prueba para software: ejemplos y tutorial. <https://es.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/>
- [4] Software Testing Bureau. (2023). Crear un buen plan de pruebas. <https://www.softwaretestingbureau.com/crear-un-buen-plan-de-pruebas/>
- [5] Solbyte Servicios informáticos. (8 de mayo de 2021). Ciclo de vida del software. Qué es, modelos y etapas. <https://www.solbyte.com/blog/ciclo-de-vida-del-software/#:~:text=Pruebas,software%20depende%C3%A1%20de%20esta%20fase.>
- [6] UNIR. (26 de agosto de 2022). La importancia de las pruebas de software. <https://www.unir.net/ingenieria/revista/pruebas-software/>