

Pruebas de software de sitio web Github

Tannia Estefanny Duarte Vargas

Ichel Alejandra Delgado Morales

Camilo Andrés Samudio Lázaro

Asesor

María Patricia Amórtegui Vargas

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería (ECBTI)

Tecnología en Desarrollo de Software

2024

Resumen

Las pruebas de software son fundamentales en el desarrollo de sistemas, ya que garantizan la calidad, funcionalidad y seguridad del producto antes de su implementación o lanzamiento. Ayudan a identificar defectos o fallos en el software que podrían causar problemas durante su uso real. Esto reduce el riesgo de errores críticos en entornos de producción. Aseguran que el producto cumpla con los estándares de calidad establecidos, incluyendo funcionalidad, rendimiento, seguridad y usabilidad. Detectar y corregir errores en etapas tempranas del desarrollo es significativamente más económico que hacerlo después del lanzamiento. Al garantizar que el software funcione según lo esperado, se mejora la experiencia del usuario final, lo que aumenta su confianza y satisfacción. Verifican que el software cumpla con las especificaciones y necesidades definidas por los clientes o partes interesadas. Al detectar problemas estructurales o de diseño, las pruebas permiten que el software sea más fácil de mantener y escalar en el futuro. Las pruebas identifican vulnerabilidades que podrían ser explotadas, protegiendo datos sensibles y evitando problemas legales o de reputación.

Palabras clave: pruebas de software, desarrollo de sistemas, calidad, funcionalidad, seguridad, producto, defectos, fallos, riesgo de errores, entornos de producción, estándares de calidad, rendimiento, usabilidad.

Abstract

Software testing is essential in system development, as it ensures the quality, functionality, and security of the product before it is deployed or released. It helps to identify defects or bugs in the software that could cause problems during actual use. This reduces the risk of critical errors in production environments. It ensures that the product meets established quality standards, including functionality, performance, security, and usability. Detecting and correcting errors in early stages of development is significantly cheaper than doing so after release. By ensuring that the software works as expected, the end-user experience is improved, increasing their confidence and satisfaction. It verifies that the software meets the specifications and needs defined by customers or stakeholders. By detecting structural or design issues, testing allows the software to be easier to maintain and scale in the future. Testing identifies vulnerabilities that could be exploited, protecting sensitive data and avoiding legal or reputational issues.

Key words: software testing, system development, quality, functionality, security, product, defects, bugs, risk of errors, production environments, quality standards, performance, usability

Tabla de Contenido

Introducción	16
Justificación.....	17
Objetivos	18
Específicos	18
Plan de Pruebas de Software	19
Elementos de Pruebas	21
Resumen de los Casos de Prueba Identificados	24
Características que no serán probadas	24
Matriz de Dependencias	25
Criterios de aceptación.....	29
Gestión de riesgos	30
Estimación de tiempos	32
Estimación de tiempos por módulo.....	32
Recursos.....	34
Requerimientos de Hardware.....	34
Requerimientos de Software	34
Herramientas de pruebas requeridas	35
Personal.....	35
Diseño de Pruebas de Software	36
Ejecución de casos de prueba.....	62
Gestión de defectos	111

Métricas.....	5
Métricas.....	114
Métricas de Ejecución.....	114
Métricas acordes a Severidad.....	115
Métricas Acordes a Tipo de Incidencia.....	117
Métricas Acordes a su Prioridad	118
Métricas por Naturaleza.....	119
Evaluación del Producto y Proveedor.....	121
Informe de Pruebas	122
Información Básica	122
Alcance	122
Características que no fueron probadas	122
Métricas de Ejecución del Proyecto.....	124
Lecciones Aprendidas	126
Conclusiones	127
Referencias Bibliográficas.....	128

Lista de tablas

Tabla 1 <i>Información general de la aplicación</i>	20
Tabla 2 <i>Elementos de pruebas</i>	22
Tabla 3 <i>Resumen de los casos de prueba</i>	24
Tabla 4 <i>Matriz de dependencias</i>	25
Tabla 5 <i>Matriz de riesgos</i>	30
Tabla 6 <i>Estimación de tiempos</i>	32
Tabla 7 <i>Estimación de tiempos por módulos</i>	33
Tabla 8 <i>Diseño de Prueba SUP_CP01_Crear_cuenta</i>	37
Tabla 9 <i>Diseño de Prueba LOG_CP02_Iniciar_sesion</i>	38
Tabla 10 <i>Diseño de Prueba REP_CP03_Crear_repositorio</i>	39
Tabla 11 <i>Diseño de Prueba REP_CP04_Eliminar_repositorio</i>	40
Tabla 12 <i>Diseño de Prueba ISS_CP05_Crear_issue</i>	41
Tabla 13 <i>Diseño de Prueba ISS_CP06_Cerrar_issue</i>	42
Tabla 14 <i>Diseño de Prueba PR_CP07_Crear_pull_request</i>	43
Tabla 15 <i>Diseño de Prueba PR_CP08_Revisar_pull_request</i>	44
Tabla 16 <i>Diseño de Prueba BR_CP09_Crear_branch</i>	45
Tabla 17 <i>Diseño de Prueba BR_CP10_Eliminar_branch</i>	46
Tabla 18 <i>Diseño de Prueba COM_CP11_Hacer_commit</i>	47
Tabla 19 <i>Diseño de Prueba COM_CP12_Revertir_commit</i>	48
Tabla 20 <i>Diseño de Prueba MER_CP13_Solicitud_merge</i>	49

Tabla 21 <i>Diseño de Prueba PRO_CP14_Crear_proyecto</i>	50
Tabla 22 <i>Diseño de Prueba PRO_CP15_Cerrar_proyecto</i>	51
Tabla 23 <i>Diseño de Prueba PRO_CP16_Eliminar_proyecto</i>	52
Tabla 24 <i>Diseño de Prueba PRO_CP17_Adicion_items_proyecto</i>	53
Tabla 25 <i>Diseño de Prueba PRO_CP18_Adicion_miembros_proyecto</i>	54
Tabla 26 <i>Diseño de Prueba NOT_CP19_Recibir_notificaciones</i>	55
Tabla 27 <i>Diseño de Prueba NOT_CP20_Leer_notificaciones</i>	56
Tabla 28 <i>Diseño de Prueba PAG_CP21_Alojar_pagina</i>	57
Tabla 29 <i>Diseño de Prueba PAG_CP21_Alojar_pagina</i>	58
Tabla 30 <i>Diseño de Prueba PAG_CP23_Personalizar_dominio</i>	59
Tabla 31 <i>Diseño de Prueba WIK_CP24_Crear_wiki</i>	60
Tabla 32 <i>Diseño de Prueba WIK_CP25_Editar_wiki</i>	61
Tabla 33 <i>Ejecución Caso de Prueba UP_CP01_Crear_cuenta</i>	62
Tabla 34 <i>Ejecución Caso de Prueba LOG_CP02_Iniciar_sesion</i>	64
Tabla 35 <i>Ejecución Caso de Prueba REP_CP03_Crear_repositorio</i>	66
Tabla 36 <i>Ejecución Caso de Prueba REP_CP04_Eliminar_repositorio</i>	68
Tabla 37 <i>Ejecución Caso de Prueba ISS_CP05_Crear_issue</i>	70
Tabla 38 <i>Ejecución Caso de Prueba ISS_CP06_Cerrar_issue</i>	72
Tabla 39 <i>Ejecución Caso de Prueba PR_CP07_Crear_pull_request</i>	74
Tabla 40 <i>Ejecución Caso de Prueba PR_CP08_Revisar_pull_request</i>	76
Tabla 41 <i>Ejecución Caso de Prueba BR_CP09_Crear_branch</i>	78

Tabla 42 <i>Ejecución Caso de Prueba BR_CP10_Eliminar_branch</i>	80
Tabla 43 <i>Ejecución Caso de Prueba COM_CP11_Hacer_commit</i>	82
Tabla 44 <i>Ejecución Caso de Prueba COM_CP12_Revertir_commit</i>	84
Tabla 45 <i>Ejecución Caso de Prueba MER_CP13_Solicitud_merge</i>	86
Tabla 46 <i>Ejecución Caso de Prueba PRO_CP14_Crear_proyecto</i>	87
Tabla 47 <i>Ejecución Caso de Prueba PRO_CP15_Cerrar_proyecto</i>	88
Tabla 48 <i>Ejecución Caso de Prueba PRO_CP16_Eliminar_proyecto</i>	90
Tabla 49 <i>Ejecución Caso de Prueba PRO_CP17_Adicion_items_proyecto</i>	92
Tabla 50 <i>Ejecución Caso de Prueba PRO_CP18_Adicion_miembros_proyecto</i>	94
Tabla 51 <i>Ejecución Caso de Prueba NOT_CP19_Recibir_notificaciones</i>	96
Tabla 52 <i>Ejecución Caso de Prueba NOT_CP20_Leer_notificaciones</i>	98
Tabla 53 <i>Ejecución Caso de Prueba PAG_CP21_Alojar_pagina</i>	100
Tabla 54 <i>Ejecución Caso de Prueba PAG_CP22_Desplegar_pagina</i>	102
Tabla 55 <i>Ejecución Caso de Prueba PAG_CP23_Personalizar_dominio</i>	104
Tabla 56 <i>Ejecución Caso de Prueba WIK_CP24_Crear_wiki</i>	106
Tabla 57 <i>Ejecución Caso de Prueba WIK_CP25_Editar_wiki</i>	108
Tabla 58 <i>Gestión de incidencias</i>	112
Tabla 59 <i>Métricas de ejecución</i>	114
Tabla 60 <i>Métricas acordes a severidad</i>	116
Tabla 61 <i>Métricas acordes al tipo de Incidencia</i>	117
Tabla 62 <i>Métricas acordes a su prioridad</i>	118

Tabla 63 <i>Métricas acordes a su naturaleza</i>	120
Tabla 64 <i>Evaluación del producto y el proveedor</i>	121
Tabla 65 <i>Métricas de ejecución del proyecto</i>	124

Lista de figuras

Figura 1 <i>Cronograma</i>	34
Figura 2 <i>Registro de correo</i>	62
Figura 3 <i>Captcha</i>	62
Figura 4 <i>Ingreso de código de verificación</i>	62
Figura 5 <i>Inicio sesión en plataforma</i>	64
Figura 6 <i>Interfaz de dashboard</i>	64
Figura 7 <i>Inicio sesión en plataforma para crear repositorio</i>	66
Figura 8 <i>Opción para crear nuevo repositorio</i>	66
Figura 9 <i>Configuración para crear nuevo repositorio</i>	66
Figura 10 <i>Verificación de datos de nuevo repositorio</i>	67
Figura 11 <i>Creación de repositorio</i>	67
Figura 12 <i>Lista repositorios</i>	67
Figura 13 <i>Inicio sesión para eliminar repositorio</i>	68
Figura 14 <i>Zona de peligro de repositorio</i>	68
Figura 15 <i>Ventana de eliminación</i>	68
Figura 16 <i>Advertencia de eliminación de repositorio</i>	69
Figura 17 <i>Confirmación de eliminación de repositorio</i>	69
Figura 18 <i>Dashboard sin el repositorio</i>	69
Figura 19 <i>Inicio sesión para Crear issue</i>	70
Figura 20 <i>Vista general de issues</i>	70

Figura 21 <i>Vista de opciones de creación de issue</i>	70
Figura 22 <i>Creación exitosa de issue</i>	71
Figura 23 <i>Inicio sesión para cerrar issue</i>	72
Figura 24 <i>Confirmación para cerrar issue</i>	72
Figura 25 <i>Issue cerrado</i>	72
Figura 26 <i>Inicio sesión para crear pull request</i>	74
Figura 27 <i>Vista general de los pull request del repositorio</i>	74
Figura 28 <i>Vista de cambios</i>	74
Figura 29 <i>Vista de pull request creado</i>	75
Figura 30 <i>Inicio sesión para revisar pull request</i>	76
Figura 31 <i>Cambios realizados en el pull request</i>	76
Figura 32 <i>Estado pull request</i>	76
Figura 33 <i>Inicio sesión para crear branch</i>	78
Figura 34 <i>Vista general de ramas del repositorio</i>	78
Figura 35 <i>Ventana de creación de rama nueva</i>	78
Figura 36 <i>Rama nueva creada</i>	79
Figura 37 <i>Inicio sesión para eliminar branch</i>	80
Figura 38 <i>Visualización de listado de branches</i>	80
Figura 39 <i>Opción de eliminar branch</i>	80
Figura 40 <i>Eliminar rama</i>	81
Figura 41 <i>Inicio sesión para realizar commit</i>	82

Figura 42 <i>Visualización de repositorio para commit</i>	82
Figura 43 <i>Creación de commit dentro de la rama unad2</i>	82
Figura 44 <i>Confirmación de creación de commit</i>	83
Figura 45 <i>Actualización de historial de commits</i>	83
Figura 46 <i>Inicio de sesión para revertir commit</i>	84
Figura 47 <i>Vista general de repositorio con commit a revertir</i>	84
Figura 48 <i>Advertencia de reversión de commit</i>	84
Figura 49 <i>Crea nuevo commit de reversión</i>	85
Figura 50 <i>Vista de commit de reversión</i>	85
Figura 51 <i>Rama destino</i>	86
Figura 52 <i>Rama fusionada</i>	86
Figura 53 <i>Crear proyecto</i>	87
Figura 54 <i>Proyecto creado</i>	87
Figura 55 <i>Menú configuración cerrar proyecto</i>	88
Figura 56 <i>Zona de peligro cerrar proyecto</i>	88
Figura 57 <i>Cerrar proyecto</i>	88
Figura 58 <i>Menú de configuración eliminar proyecto</i>	90
Figura 59 <i>Zona de peligro eliminar proyecto</i>	90
Figura 60 <i>Eliminar proyecto</i>	90
Figura 61 <i>Inicio de sesión para adicionar proyectos</i>	92
Figura 62 <i>Página inicial de repositorios de usuario</i>	92

Figura 63 <i>Vista general de proyecto</i>	92
Figura 64 <i>Vista general de proyecto seleccionado</i>	93
Figura 65 <i>Agregar datos nuevo ítem</i>	93
Figura 66 <i>Item creado</i>	93
Figura 67 <i>Inicio sesión para adición de miembros</i>	94
Figura 68 <i>Página inicial adición miembros</i>	94
Figura 69 <i>Vista general de página repositorio</i>	94
Figura 70 <i>Adición de datos de nuevo miembro</i>	95
Figura 71 <i>Adición de nuevo miembro</i>	95
Figura 72 <i>Lista de miembros de proyectos</i>	95
Figura 73 <i>Iniciar sesión para recibir notificaciones</i>	96
Figura 74 <i>Plataforma inicial para recibir notificaciones</i>	96
Figura 75 <i>Icono notificaciones</i>	96
Figura 76 <i>Iniciar sesión para leer notificaciones</i>	98
Figura 77 <i>Página inicial para leer notificaciones</i>	98
Figura 78 <i>Icono notificación para leer notificaciones</i>	98
Figura 79 <i>Visualización de las notificaciones</i>	99
Figura 80 <i>Visualización de detalle notificación</i>	99
Figura 81 <i>Iniciar sesión para alojar página</i>	100
Figura 82 <i>Página inicial de repositorios para alojar</i>	100
Figura 83 <i>Visualización de proyectos para alojar</i>	100

Figura 84 <i>Configuración repositorios de alojamiento</i>	101
Figura 85 <i>Guardar alojamiento del proyecto</i>	101
Figura 86 <i>Iniciar sesión para desplegar proyecto</i>	102
Figura 87 <i>Página inicial para desplegar proyecto</i>	102
Figura 88 <i>Visualización de proyectos a desplegar</i>	102
Figura 89 <i>Configuración de despliegue de proyecto</i>	103
Figura 90 <i>Guarda configuración de despliegue</i>	103
Figura 91 <i>Configuración final de despliegue</i>	103
Figura 92 <i>Iniciar sesión para personalizar dominio</i>	104
Figura 93 <i>Página inicial para personalizar dominio</i>	104
Figura 94 <i>Visualización de proyectos para personalizar</i>	104
Figura 95 <i>Cambio dominio de proyecto</i>	105
Figura 96 <i>Guarda configuración de dominio</i>	105
Figura 97 <i>Página con nuevo dominio</i>	105
Figura 98 <i>Iniciar sesión para crear wiki</i>	106
Figura 99 <i>Página inicial de proyectos para crear wiki</i>	106
Figura 100 <i>Visualización de proyectos para crear wiki</i>	106
Figura 101 <i>Visualización de sección de wiki</i>	107
Figura 102 <i>Creación de página de wiki</i>	107
Figura 103 <i>Visualización de página de wiki creada</i>	107
Figura 104 <i>Iniciar sesión para editar wiki</i>	108

Figura 105 <i>Página inicial de proyectos para editar wiki</i>	108
Figura 106 <i>Visualización de proyectos para editar wiki</i>	108
Figura 107 <i>Visualización de módulo wiki</i>	109
Figura 108 <i>Ventana de edición de wiki</i>	109
Figura 109 <i>Edición de opciones de wiki</i>	109
Figura 110 <i>Confirmación de edición de wiki</i>	109
Figura 111 <i>Métricas de ejecución</i>	115
Figura 112 <i>Métricas de severidad</i>	116
Figura 113 <i>Métricas por tipo de incidencia</i>	118
Figura 114 <i>Métricas acordes a su prioridad</i>	119
Figura 115 <i>Métricas por naturaleza</i>	120

Introducción

Las pruebas de software representan un componente esencial en el ciclo de desarrollo de aplicaciones y sistemas, ya que permiten garantizar la calidad, el rendimiento y la seguridad del producto final. En un contexto donde la demanda de software confiable y eficiente es cada vez mayor, resulta fundamental implementar procesos estandarizados que aseguren la detección temprana de errores y la mejora continua del producto. Estas prácticas no solo contribuyen a la satisfacción del usuario final, sino que también fortalecen la reputación de las empresas y desarrolladores en un mercado altamente competitivo.

Este trabajo se enfoca en la aplicación de pruebas de software para la plataforma GitHub, una de las herramientas más utilizadas en el desarrollo colaborativo de proyectos. Todo esto, basándose en normativas y estándares reconocidos, como la NTC 5421, que proporcionan un marco de referencia para la ejecución de pruebas sistemáticas y efectivas. A través de este enfoque, se busca no solo validar la funcionalidad de la plataforma, sino también identificar áreas de mejora que contribuyan a su evolución y adaptación a las necesidades de los usuarios.

Justificación

La importancia de realizar pruebas de software en GitHub radica en que estas son fundamentales para garantizar la calidad, funcionalidad y confiabilidad del sistema, reduciendo los riesgos y costos asociados con posibles errores. A través de estas pruebas, se verifica que el software cumpla con los requisitos técnicos y funcionales establecidos, identificando defectos antes de que lleguen a la producción, lo cual resulta más económico que corregirlos en fases posteriores. Además, las pruebas previenen fallos que podrían impactar negativamente al usuario final, a la organización o al mercado.

Las pruebas de software garantizan que el software sea intuitivo, eficiente y que satisfaga las expectativas de los usuarios. Un software bien probado tiene menos defectos, lo que facilita su mantenimiento y la integración de nuevas funcionalidades en plataformas como GitHub.

Objetivos

General

Garantizar la calidad, funcionalidad y confiabilidad de la plataforma GitHub mediante la implementación de pruebas de software, con el fin de identificar defectos, sugerencias o consideraciones y asegurar que el producto satisfaga las expectativas de los usuarios.

Específicos

Implementar pruebas automatizadas y manuales para verificar que las funcionalidades clave de GitHub operen correctamente, identificando y corrigiendo defectos antes de su implementación en producción.

Evaluar la usabilidad e interfaz de usuario de GitHub mediante pruebas de interacción para asegurar que la plataforma sea intuitiva y satisfaga las expectativas de los usuarios finales.

Analizar las métricas de errores de ejecución, severidad, tipos de incidencia, prioridad y naturaleza para identificar patrones y áreas críticas en la plataforma GitHub, con el fin de optimizar la gestión de incidencias y priorizar las mejoras según su impacto en la experiencia del usuario.

Plan de Pruebas de Software

Un plan de pruebas de software es un documento esencial que define el enfoque, los recursos y el cronograma necesarios para realizar las pruebas de un sistema o aplicación. Según Pressman (2014), el plan de pruebas describe los objetivos de las pruebas, las actividades a realizar, las responsabilidades de los involucrados y los criterios de aceptación del producto. Este plan garantiza que el proceso de pruebas sea estructurado y que se cubran todos los aspectos críticos del sistema para minimizar errores y asegurar la calidad. A su vez, Myers (2011) enfatiza que un plan de pruebas eficaz considera tanto las pruebas funcionales como las no funcionales, asegurando que se evalúen tanto el rendimiento como la usabilidad del sistema.

En este contexto, GitHub, como una de las plataformas de desarrollo colaborativo más utilizadas a nivel global, representa un caso de estudio ideal para la aplicación de un plan de pruebas riguroso. Con funcionalidades como repositorios, solicitudes de extracción (pull requests) e integraciones con herramientas de CI/CD, GitHub se ha convertido en un estándar para el desarrollo de software moderno. Además de su impacto técnico, GitHub ha fomentado una comunidad global que contribuyen a proyectos públicos y colaboran en soluciones innovadoras.

Para garantizar la calidad y funcionalidad de una plataforma tan crítica como GitHub, es fundamental llevar a cabo pruebas de software exhaustivas que permitan identificar y corregir errores, optimizar el rendimiento y asegurar una experiencia de usuario satisfactoria. A continuación, en la Tabla 1, se presenta el equipo responsable de la ejecución de estas pruebas, detallando los roles y responsabilidades de cada integrante.

Tabla 1*Información general de la aplicación*

Nombre de la aplicación a evaluar:	Github
Grupo de trabajo:	Líder:
Tannia Estefanny Duarte Vargas	Tannia Estefanny Duarte Vargas
Ichel Alejandra Delgado Morales	Funciones:
Camilo Andres Samudio Lazaro	Asume la responsabilidad general de la revisión. Decide quiénes estarán involucrados y organiza cuándo y dónde se llevará a cabo.
	Probadores:
	Nombre de probador 1: Ichel Alejandra Delgado Morales
	Nombre de probador 2: Camilo Andres Samudio Lazaro
	Funciones:
	Realizar la prueba inicial y la prueba de confirmación final.
	Depuración y la prueba de componente.
	Revisión de los requisitos o en el refinamiento de historias de usuario.
	Trabajar en estrecha colaboración con los diseñadores y desarrolladores de sistemas mientras se diseña el sistema.
	Verificar y validar el software antes de liberarlo.
Fecha de inicio de pruebas:	24/09/2024
Fecha fin de pruebas:	30/09/2024

Nota. Información general de la aplicación y equipo de trabajo. *Fuente.* Autoría propia

Elementos de Pruebas

Los elementos de prueba representan los componentes específicos del software que serán evaluados durante el proceso de pruebas. Estos elementos pueden incluir módulos, funcionalidades, interfaces, bases de datos, y cualquier otro aspecto que interactúe dentro del sistema. En el caso de GitHub, una plataforma compleja y multifuncional, los principales elementos de prueba abarcan una amplia gama de características críticas para su operación. Entre estos se encuentran el sistema de gestión de repositorios, que permite a los usuarios almacenar, organizar y versionar su código; las herramientas de colaboración, como los issues (para seguimiento de problemas) y las pull requests (para revisión y fusión de código); la funcionalidad de seguimiento de versiones, que garantiza el control preciso de los cambios en el código; y los servicios de integración continua (CI), que automatizan la construcción y prueba del software.

Evaluar estos elementos permite garantizar que el sistema cumpla con los requisitos funcionales y no funcionales establecidos, y que ofrezca una experiencia de usuario estable y eficiente. La evaluación exhaustiva de estos elementos permite asegurar que GitHub cumpla con los requisitos establecidos, ofreciendo a los usuarios una plataforma estable, confiable y de alto rendimiento. Este enfoque integral en las pruebas no solo minimiza los riesgos de fallos, sino que también contribuye a la mejora continua de la plataforma, adaptándose a las necesidades cambiantes de la comunidad de desarrolladores. A continuación, en la Tabla 1, se presentan los elementos de prueba para este proceso:

Tabla 2*Elementos de pruebas*

#	Módulo	Nombre del caso de prueba	Descripción	Prioridad	Responsable
1	Sign up	SUP_CP01_Crear_cuenta	Verifica que el usuario pueda crear una cuenta para iniciar sesión.	Alta	Camilo Andres Samudio Lazaro
2	Log in	LOG_CP02_Iniciar_sesion	Verifica que el usuario pueda iniciar sesión.	Alta	Camilo Andres Samudio Lazaro
3	Repositories	REP_CP03_Crear_repositorio	Verifica que el usuario pueda crear un nuevo repositorio.	Alta	Tannia Estefanny Duarte Vargas
4	Repositories	REP_CP04_Eliminar_repositorio	Comprueba si el usuario puede eliminar un repositorio existente.	Media	Tannia Estefanny Duarte Vargas
5	Issues	ISS_CP05_Crear_issue	Verifica la creación de issues para seguimiento de tareas.	Alta	Tannia Estefanny Duarte Vargas
6	Issues	ISS_CP06_Cerrar_issue	Evalúa si el usuario puede cerrar un issue correctamente.	Media	Tannia Estefanny Duarte Vargas
7	Pull Requests	PR_CP07_Crear_pull_request	Comprueba la creación de una nueva pull request	Alta	Tannia Estefanny Duarte Vargas
8	Pull Requests	PR_CP08_Revisar_pull_request	Evalúa si es posible revisar y aprobar pull requests	Alta	Tannia Estefanny Duarte Vargas
9	Branches	BR_CP09_Crear_branch	Verifica que el usuario puede crear un nuevo branch	Alta	Tannia Estefanny Duarte Vargas
10	Branches	BR_CP10_Eliminar_branch	Comprueba si se puede eliminar un branch	Media	Tannia Estefanny Duarte Vargas
11	Commits	COM_CP11_Hacer_commit	Verifica la funcionalidad para hacer un commit con cambios	Alta	Tannia Estefanny Duarte Vargas
12	Commits	COM_CP12_Revertir_commit	Evalúa la capacidad de revertir un commit	Media	Tannia Estefanny Duarte Vargas

13	Merge	MER_CP13_Solicitud_merge	Comprueba la posibilidad de realizar una solicitud de merge.	Alta	Camilo Andres Samudio Lazaro
14	Projects	PRO_CP14_Crear_proyecto	Comprueba la creación de un nuevo proyecto	Alta	Camilo Andres Samudio Lazaro
15	Projects	PRO_CP15_Cerrar_proyecto	Comprueba el cierre de un proyecto	Alta	Camilo Andres Samudio Lazaro
16	Projects	PRO_CP16_Eliminar_proyecto	Comprueba la eliminación de un proyecto	Alta	Camilo Andres Samudio Lazaro
18	Repositories	PRO_CP18_Adicion_miembros_proyecto	Comprueba la posibilidad de agregar nuevos miembros a los proyectos	Alta	Ichel Alejandra Delgado Morales
19	Notifications	NOT_CP19_Recibir_notificaciones	Verifica que el usuario pueda recibir notificaciones	Alta	Ichel Alejandra Delgado Morales
20	Notifications	NOT_CP20_Leer_notificaciones	Verifica que el usuario pueda leer las notificaciones que recibe	Alta	Ichel Alejandra Delgado Morales
21	Pages	PAG_CP21_Alojar_pagina	Verifica que el usuario pueda alojar una página en GitHub.	Alta	Ichel Alejandra Delgado Morales
22	Pages	PAG_CP22_Desplegar_pagina	Verifica que el usuario pueda desplegar una página en GitHub.	Alta	Ichel Alejandra Delgado Morales
23	Pages	PAG_CP23_Personalizar_dominio	Verifica que el usuario pueda asignar un dominio personalizado a una página alojada en GitHub.	Alta	Ichel Alejandra Delgado Morales
24	Wiki	WIK_CP24_Crear_wiki	Comprueba que el usuario pueda crear una Wiki en un repositorio existente.	Alta	Ichel Alejandra Delgado Morales
25	Wiki	WIK_CP25_Editar_wiki	Comprueba que el usuario pueda editar una Wiki en un repositorio existente.	Alta	Ichel Alejandra Delgado Morales

Nota. Se muestran y describen los 25 elementos de prueba para el sitio web GitHub. *Fuente.* Autoría propia

Resumen de los Casos de Prueba Identificados

Tabla 3

Resumen de los casos de prueba

Tipo de CP	Total
Prioridad Alta	21
Prioridad Media:	4
Prioridad Baja:	0
Total CP:	25

Nota. Se el resumen de los totales de los casos de prueba por prioridad. *Fuente.* Autoría propia

Características que no serán probadas

Las características que no serán probadas son: Appearance and profile settings, Access, Security, Integrations, Archives y Developer settings.

Tannia	07	X	X	X		
Estefanny						
Duarte						
Vargas						
Tannia	08	X	X	X	X	
Estefanny						
Duarte						
Vargas						
Tannia	09	X	X	X		
Estefanny						
Duarte						
Vargas						
Tannia	10	X	X	X	X	
Estefanny						
Duarte						
Vargas						
Tannia	11	X	X	X	X	
Estefanny						
Duarte Vargas						
Tannia	12	X	X	X	X	X
Estefanny						
Duarte						
Vargas						
Camilo	13	X	X	X	X	
Andres						
Samudio						
Lazaro						
Camilo						
Andres	14	X	X	X		
Samudio						
Lazaro						

Ichel Alejandra Delgado Morales	23	X	X	X			X	X
Ichel Alejandra Delgado Morales	24	X	X	X				
Ichel Alejandra Delgado Morales	25	X	X	X				X

Nota. Se el resumen las relaciones de dependencia de los casos de prueba propuestos. *Fuente.* Autoría propia

Metodología de pruebas

La ejecución del proyecto que cubre el presente Plan de Pruebas se realiza en 3 etapas: Planeación, diseño y ejecución. En la etapa de planeación se construye el plan de pruebas con el fin de identificar objetivos, alcance y componentes a probar, así estimar el esfuerzo y los recursos del proceso.

En la etapa de Diseño se realiza el diseño de cada uno de los casos de prueba identificados en la planeación. En la etapa de Ejecución se ejecutan los casos de prueba para verificar si son exitosos o no. Todo esto es soportado por una adecuada gestión, seguimiento y control de proyectos, el aseguramiento de calidad de procesos y productos y la gestión de la configuración.

Criterios de aceptación

El proceso de pruebas funcionales se da por terminado una vez que:

Se han ejecutado el 85% de los casos de prueba diseñados para este proyecto y su resultado ha sido exitoso.

No debe haber defectos de prioridad alta o crítica pendientes de resolución en el sistema. Todos los defectos encontrados deben estar documentados y, al menos, el 90% de los defectos de prioridad media deben estar resueltos o contar con un plan de mitigación.

Todos los defectos críticos y de alta prioridad encontrados durante las pruebas han sido corregidos y verificado

Gestión de riesgos

Es importante identificar los riesgos asociados al proyecto, a continuación, se detallan los riesgos del proyecto:

Tabla 5

Matriz de riesgos

Matriz de Riesgos										
Proyecto: Pruebas de software de Github										
Fecha inicio: 24/09/2024										
Fecha fin: 30/09/2024										
# Riesgo	Tipo	Riesgo			Impacto	Probabilidad	Evaluación		Respuesta	Autor
		Origen	Resultado	Señal			Nivel	Valor		
1	Tiempo, costo, alcance, Calidad	El sistema podría no permitir la creación de cuentas nuevas debido a errores en la validación de datos.	Los usuarios no podrán acceder al sistema.	Mensajes de error de credenciales. No recepción de correo para validar cuenta.	B	B	Apreciable	3	Implementar validaciones exhaustivas en los campos de entrada y realizar pruebas unitarias y de integración para garantizar la funcionalidad. Además, establecer un proceso de monitoreo y registro de errores para identificar y corregir fallos en la validación de datos de manera oportuna.	Ichel Alejandra Delgado Morales
2	Riesgo	latencia en las	Lentitud	Lentitud en	B	B	Importante	4	Verificación del	Camilo

	Operativo	bases de datos.	en todos los procesos a nivel general.	cargar cualquier página dentro de la plataforma, también para realizar cualquier proceso.					servidor de las bases de datos, ver si hay procesos saturados o queries que están generando conflicto.	Andres Samudio Lazaro
3	Riesgo de seguridad	Acceso no autorizado a la cuenta de usuario.	Pérdida de datos sensibles y violación a la privacidad del usuario.	Actividades inusuales en las cuentas de usuario, como intentos fallidos de inicio de sesión.	A	A	Crítico	5	Implementar MFA, auditorías de seguridad y capacitación en protección de credenciales.	Tannia Estefanny Duarte Vargas
4	Riesgo de recursos	Falta de personal cualificado para manejar problemas técnicos.	Retrasos en la resolución de problemas y afectación de la calidad del servicio.	Aumento en el tiempo de respuesta para resolver incidencias.	A	A	Importante	3	Planificar la formación continua del personal y contar con un equipo de soporte técnico que pueda atender incidencias de manera eficiente. Considerar la posibilidad de contratar personal adicional o subcontratar para cubrir picos de trabajo.	Tannia Estefanny Duarte Vargas

Nota. Se resumen los riesgos encontrados, su señal, impacto, probabilidad, evaluación y respuesta. *Fuente.* Autoría propia

Estimación de tiempos

La técnica de estimación de tiempos es a juicio de expertos; con la prueba de humo realizada, los casos de prueba identificados, la matriz de dependencias y de riesgos, se debe proyectar el tiempo que el probador invertirá al diseñar, ejecutar y documentar los resultados de los casos de prueba.

Tabla 6

Estimación de tiempos

Etapa Impacto	Diseño			Ejecución			Documentación		
	Alta	Media	Baja	Alta	Media	Baja	Alta	Media	Baja
Peso en Minutos	120	90	60	100	70	50	120	90	60
No. De Casos De Prueba	21	4	0	21	4	0	21	4	0
Subtotal	2520	360	0	2100	280	0	2520	360	0
Total	2880 minutos 48 horas			2380 minutos 40 horas			2880 minutos 48 horas		

Nota. Se resumen los tiempos estimados para cada fase del plan de pruebas. *Fuente.* Autoría propia.

Estimación de tiempos por módulo

En la tabla 7, se detalla la estimación de tiempos por cada módulo para estimar el esfuerzo de cada integrante del equipo de pruebas. Para ello tenga en cuenta los datos de las anteriores tablas y diligencia la siguiente:

Tabla 7*Estimación de tiempos por módulos*

Responsables	Módulos	Diseño	Ejecución	Documentación
Ichel Alejandra Delgado Morales, Camilo Andres Samudio	Projects	600 minutos	500 minutos	600 minutos
Ichel Alejandra Delgado Morales	Notifications	240 minutos	200 minutos	240 minutos
Ichel Alejandra Delgado Morales	Pages	360 minutos	300 minutos	360 minutos
Ichel Alejandra Delgado Morales	Wiki	240 minutos	200 minutos	240 minutos
Camilo Andres Samudio	Creación usuario, login	240 minutos	200 minutos	240 minutos
Camilo Andres Samudio	Merge	120 minutos	100 minutos	120 minutos
Tannia Estefanny Duarte Vargas	Repositorio	480 minutos	240 minutos	240 minutos
Tannia Estefanny Duarte Vargas	Issue	240 minutos	120 minutos	120 minutos
Tannia Estefanny Duarte Vargas	Pull_request	240 minutos	100 minutos	120 minutos
Tannia Estefanny Duarte Vargas	Branches	120 minutos	120 minutos	100 minutos
Tannia Estefanny Duarte Vargas	Commits	240 minutos	100 minutos	120 minutos
	Total (Horas).	52	36.3	41.7

Nota. Se resumen los tiempos estimados por módulo para cada fase del plan de pruebas.

Fuente. Autoría propia

Figura 1

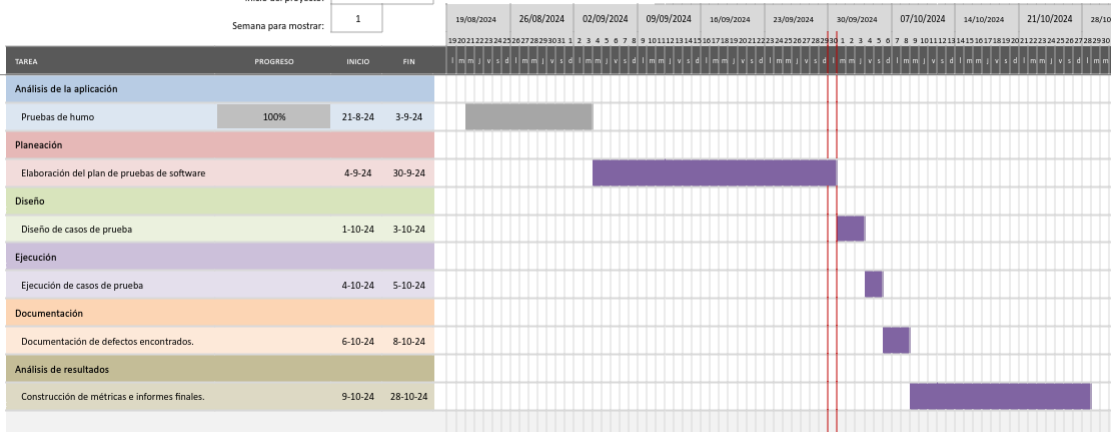
Cronograma

Plan de Pruebas de Software: GitHub

Tannia Estefanny Duarte Vargas

Inicio del proyecto: mi., 8/21/2024

Semana para mostrar: 1



Fuente. Autoría propia

Recursos

Se listan los elementos necesarios para desarrollar las pruebas:

Requerimientos de Hardware

Computadoras/Estaciones de trabajo. Mínimo de 16 GB de RAM, procesador de al menos 4 núcleos (i5 o equivalente) y almacenamiento SSD de al menos 256 GB.

Servidor de Pruebas. Al menos 32 GB de RAM, procesador de alto rendimiento (i7 o equivalente) y almacenamiento SSD de al menos 1 TB.

Red. Conexión a internet de alta velocidad, router y switch para la conexión en red local.

Requerimientos de Software

Sistema Operativo. Windows 10/11, Linux (Ubuntu, CentOS), o macOS (dependiendo de las herramientas utilizadas).

Lenguajes de Programación. Java, JavaScript o el lenguaje que se utilice para las pruebas.

Entornos de Desarrollo. IDEs como Visual Studio Code.

Sistemas de Gestión de Versiones. Git y GitHub para la gestión de repositorios y control de versiones.

Herramientas de pruebas requeridas

Herramientas de Pruebas Automatizadas. Para ejecutar pruebas automáticamente.

Herramientas de Pruebas de Carga. Evaluación del rendimiento de las aplicaciones.

Herramientas de Gestión de Pruebas. Organización y seguimiento del progreso de pruebas.

Herramientas de Análisis de Código. Para asegurar la calidad del código en los repositorios.

Personal

Gerente de Proyecto. Coordinación del equipo de pruebas y desarrollo.

Desarrolladores. Responsables del código y casos de prueba

Ingenieros de Pruebas. Ejecución de pruebas y reporte de errores

Analista de Calidad. Revisión de resultados y gestión de cambios.

Soporte Técnico. Resolución de problemas técnicos.

Diseño de Pruebas de Software

El diseño de pruebas de software es un proceso crítico en el desarrollo de software que implica la creación de casos de prueba efectivos para evaluar la calidad, el rendimiento y la funcionalidad del software antes de su implementación o lanzamiento. Un diseño de pruebas bien estructurado permite detectar defectos en etapas tempranas, reduciendo costos y mejorando la satisfacción del usuario final.

Este proceso consta de los siguientes elementos clave:

Determinar los objetivos

Establecer los objetivos y alcance de las pruebas, incluyendo los requisitos funcionales y no funcionales del software.

Estrategia de Pruebas

Definir la estrategia general de pruebas, incluyendo los tipos de pruebas (manuales o automatizadas), criterios de entrada/salida y requisitos de cobertura de pruebas.

Casos de Prueba

Diseñar casos de prueba específicos que evalúen la funcionalidad del software.

Datos de Prueba

Definir los datos de prueba que se utilizarán durante las pruebas

Criterios de Aceptación

Establecer los criterios de aceptación para evaluar el éxito de las pruebas.

Gestión de Defectos

Planificar la gestión y documentación de defectos encontrados durante la prueba.

A continuación, se muestran las tablas con los diseños de los casos de prueba propuestos, detallando los escenarios y sus respectivos criterios de validación:

Tabla 8*Diseño de Prueba SUP_CP01_Crear_cuenta*

Datos de la prueba	
Nombre de CP	SUP_CP01_Crear_cuenta
Responsable:	Camilo Samudio
Módulo:	Sign up
Casos de prueba	
Precondiciones de la Prueba	El usuario puede ingresar a la plataforma El usuario puede crear una cuenta
Paso a paso de la prueba	
El usuario selecciona la opción Sign up	
El sistema muestra una casilla para ingresar el email con el cual se realiza la verificación del email	
El usuario introduce contraseña, Usarname.	
El sistema verifica que uno no sea un robot.	
Insertar código de verificación.	
Postcondiciones de la prueba:	Usuario ya está disponible para uso
Criterios de aceptación	El usuario se crea exitosamente. Si el nombre del usuario ya existe, el sistema muestra un mensaje de error y no permite la creación, pero da la opción de un usuario similar. Si la contraseña está en una lista de contraseñas comprometidas hay que cambiar a otra.

Nota. Información de CP SUP_CP01_Crear_cuenta. *Fuente.* Autoría propia

Tabla 9*Diseño de Prueba LOG_CP02_Iniciar_sesion*

Datos de la prueba	
Nombre de CP	LOG_CP02_Iniciar_sesion
Responsable:	Camilo Samudio
Módulo:	Log in
Casos de la prueba	
Precondiciones de la Prueba	El usuario puede ingresar a la plataforma
	El usuario puede ingresar el username y password de forma correcta.
Paso a Paso de la prueba	
	El usuario ingresa el username y la contraseña.
	Se le da al botón Sign in.
	Ingresa al dashboard de la aplicación.
Postcondiciones de la prueba:	El usuario está listo para utilizar Github.
Criterios de aceptación	Ingresar correctamente el usuario y la contraseña.

Nota. Información de CP LOG_CP02_Iniciar_sesion. *Fuente.* Autoría propia

Tabla 10*Diseño de Prueba REP_CP03_Crear_repositorio*

Datos de prueba	
Nombre de CP	REP_CP03_Crear_repositorio
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Repositorios
Casos de prueba	
Precondiciones de la Prueba	<p>El usuario debe haber iniciado sesión en el sistema.</p> <p>El usuario debe tener permisos para crear repositorios.</p>
Paso a Paso de la prueba	
<p>El usuario selecciona la opción Crear nuevo repositorio en el módulo de Repositorios.</p> <p>El sistema muestra un formulario en el que se solicita el nombre y la descripción del repositorio.</p> <p>El usuario introduce el nombre y la descripción y selecciona la opción Crear.</p> <p>El sistema verifica que el nombre del repositorio sea único en el espacio de trabajo del usuario.</p> <p>El sistema crea el repositorio y muestra una notificación confirmando que el repositorio ha sido creado exitosamente.</p>	
Postcondiciones de la prueba:	El nuevo repositorio aparece en la lista de repositorios del usuario y está disponible para uso.
Criterios de aceptación	<p>El repositorio se crea exitosamente y es visible en la lista de repositorios.</p> <p>Si el nombre del repositorio ya existe, el sistema muestra un mensaje de error y no permite la creación.</p>

Nota. Información de CP REP_CP03_Crear_repositorio. *Fuente.* Autoría propia

Tabla 11*Diseño de Prueba REP_CP04_Eliminar_repositorio*

Datos de prueba	
Nombre de CP	REP_CP04_Eliminar_repositorio
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Repositorios
Casos de prueba	
Precondiciones de la Prueba	El usuario debe haber iniciado sesión.
	El usuario debe tener al menos un repositorio existente.
	El usuario debe tener permisos de eliminación sobre el repositorio.
Paso a Paso de la prueba	
El usuario selecciona un repositorio de la lista y elige la opción Eliminar.	
El sistema muestra una advertencia de confirmación que indica que esta acción no se puede deshacer.	
El usuario confirma la eliminación del repositorio.	
El sistema elimina el repositorio y muestra una notificación de éxito.	
Postcondiciones de la prueba:	El repositorio eliminado ya no aparece en la lista de repositorios del usuario.
Criterios de aceptación	El repositorio se elimina exitosamente y desaparece de la lista de repositorios.
	Si el usuario intenta eliminar un repositorio inexistente, el sistema debe mostrar un mensaje de error.

Nota. Información de CP REP_CP04_Eliminar_repositorio. *Fuente.* Autoría propia.

Tabla 12*Diseño de Prueba ISS_CP05_Crear_issue*

Datos de prueba	
Nombre de CP	ISS_CP05_Crear_issue
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Issues
Casos de prueba	
Precondiciones de la Prueba	<p>El usuario debe haber iniciado sesión en el sistema.</p> <p>El usuario debe tener permisos en un repositorio donde crear un issue.</p>
Paso a Paso de la prueba	
	<p>El usuario accede a la sección de Issues en el repositorio y selecciona Crear nuevo issue.</p> <p>El sistema presenta un formulario para ingresar el título, la descripción y los posibles tags o asignados al issue.</p> <p>El usuario completa los campos requeridos y selecciona Crear.</p> <p>El sistema procesa la solicitud y crea el issue, mostrando el nuevo issue en la lista de issues del repositorio.</p>
Postcondiciones de la prueba:	El issue creado aparece en la lista de issues del repositorio y está disponible para su seguimiento.
Criterios de aceptación	<p>El issue se crea correctamente y es visible en la lista de issues.</p> <p>Si el título del issue está vacío o es duplicado, el sistema muestra un mensaje de error y no permite la creación.</p>

Nota. Información de CP ISS_CP05_Crear_issue. *Fuente.* Autoría propia.

Tabla 13*Diseño de Pruebas ISS_CP06_Cerrar_issue*

Datos de la prueba	
Nombre de CP	ISS_CP06_Cerrar_issue
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Issues
Casos de prueba	
Precondiciones de la Prueba	El usuario debe haber iniciado sesión y tener permisos en el repositorio. Debe haber al menos un issue abierto en el repositorio.
Paso a Paso de la prueba:	
	El usuario selecciona un issue abierto en la lista y elige la opción "Cerrar". El sistema solicita confirmación para cerrar el issue. El usuario confirma el cierre. El sistema cambia el estado del issue a "Cerrado" y lo muestra en la lista de issues cerrados.
Postcondiciones de la prueba:	El issue aparece en la lista de issues cerrados y no puede ser editado.
Criterios de aceptación	El issue se cierra exitosamente y cambia de estado a Cerrado. Si el issue ya está cerrado, el sistema no permite realizar la acción de cierre nuevamente.

Nota. Información de CP ISS_CP06_Cerrar_issue. *Fuente.* Autoría propia.

Tabla 14*Diseño de Prueba PR_CP07_Crear_pull_request*

Datos de la prueba	
Nombre de CP	PR_CP07_Crear_pull_request
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Pull Requests
Casos de prueba	
Precondiciones de la Prueba	El usuario debe tener cambios en una rama que no está en la rama principal. El usuario debe estar autenticado.
Paso a Paso de la prueba:	
El usuario accede a la sección de Pull Requests y selecciona Crear nueva Pull Request.	
El sistema muestra un formulario para describir los cambios, asignar revisores y añadir comentarios.	
El usuario completa los campos y selecciona Crear Pull Request.	
El sistema valida los cambios y crea la Pull Request, mostrando una confirmación de éxito.	
Postcondiciones de la prueba:	La Pull Request aparece en la lista para su revisión y es visible para los revisores asignados.
Criterios de aceptación	La Pull Request se crea correctamente y es visible en la lista. Si el usuario no tiene permisos, el sistema muestra un mensaje de error y no permite la creación.

Nota. Información de CP PR_CP07_Crear_pull_request. *Fuente.* Autoría propia.

Tabla 15*Diseño de Prueba PR_CP08_Revisar_pull_request*

Datos de la prueba	
Nombre de CP	PR_CP08_Revisar_pull_request
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Pull Requests
Casos de prueba	
Precondiciones de la Prueba	El usuario debe tener acceso a una Pull Request creada en el sistema. La Pull Request debe estar en estado pendiente de revisión.
Paso a Paso de la prueba:	
El usuario selecciona una Pull Request de la lista y elige la opción Revisar.	
El sistema muestra los cambios realizados en el Pull Request.	
El usuario revisa los cambios y selecciona Aprobar o Rechazar.	
El sistema procesa la decisión y muestra una confirmación.	
Postcondiciones de la prueba:	La Pull Request muestra el estado de aprobación o rechazo, y los cambios se pueden fusionar si son aprobados.
Criterios de aceptación	La Pull Request se revisa y cambia de estado según la acción tomada. Si la revisión no se completa, el sistema muestra el estado como pendiente.

Nota. Información de CP PR_CP08_Revisar_pull_request. *Fuente.* Autoría propia.

Tabla 16*Diseño de Prueba BR_CP09_Crear_branch*

Datos de la prueba	
Nombre de CP	BR_CP09_Crear_branch
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Branches
Casos de prueba	
Precondiciones de la Prueba	El usuario debe estar autenticado.
	El usuario debe tener permisos de creación en el repositorio.
Paso a Paso de la prueba:	
	El usuario selecciona la opción Crear nueva rama en el repositorio. El sistema solicita el nombre de la nueva rama.
	El usuario ingresa el nombre y confirma.
	El sistema crea la nueva rama y la muestra en la lista de ramas.
Postcondiciones de la prueba:	La nueva rama aparece en la lista de ramas y puede ser utilizada para hacer commits.
Criterios de aceptación	La rama se crea exitosamente y es visible en el repositorio.
	Si el nombre de la rama ya existe, el sistema muestra un mensaje de error.

Nota. Información de CP BR_CP09_Crear_branch. *Fuente.* Autoría propia.

Tabla 17*Diseño de Pruebas de Software BR_CP10_Eliminar_branch*

Datos de la prueba	
Nombre de CP	BR_CP10_Eliminar_branch
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Branches
Casos de prueba	
Precondiciones de la Prueba	El usuario debe tener al menos una rama creada. El usuario debe tener permisos de eliminación.
Paso a Paso de la prueba:	
El usuario selecciona una rama de la lista y elige Eliminar. El sistema solicita confirmación para eliminar la rama.	
El usuario confirma.	
El sistema elimina la rama y la retira de la lista.	
Postcondiciones de la prueba:	La rama eliminada desaparece de la lista de ramas.
Criterios de aceptación	La rama se elimina correctamente. Si la rama es la principal, el sistema no permite su eliminación.

Nota. Información de CP BR_CP10_Eliminar_branch. *Fuente.* Autoría propia.

Tabla 18*Diseño de Prueba COM_CP11_Hacer_commit*

Datos de la prueba	
Nombre de CP	COM_CP11_Hacer_commit
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Commits
Casos de prueba	
Precondiciones de la Prueba	<p>El usuario debe estar autenticado.</p> <p>El usuario debe tener una rama en la que se puedan realizar cambios.</p> <p>Debe haber al menos un cambio pendiente en los archivos del repositorio.</p>
Paso a Paso de la prueba:	
<p>El usuario realiza cambios en los archivos del repositorio.</p> <p>El usuario accede a la sección de commits y selecciona la opción Hacer commit. El sistema muestra un formulario para ingresar el mensaje del commit.</p> <p>El usuario ingresa un mensaje descriptivo para el commit y confirma.</p> <p>El sistema guarda el commit, registrando los cambios realizados y el mensaje proporcionado.</p> <p>El sistema muestra una notificación de éxito para confirmar que el commit se ha realizado correctamente.</p>	
Postcondiciones de la prueba:	El commit aparece en el historial de commits de la rama seleccionada y está disponible para revisión.
Criterios de aceptación	<p>El commit se realiza exitosamente y aparece en el historial.</p> <p>El mensaje del commit es visible y descriptivo.</p> <p>Si no hay cambios en los archivos, el sistema muestra un mensaje de error y no permite hacer el commit.</p>

Nota. Información de CP COM_CP11_Hacer_commit. *Fuente.* Autoría propia.

Tabla 19*Diseño de Prueba COM_CP12_Revertir_commit*

Datos de la prueba	
Nombre de CP	COM_CP12_Revertir_commit
Responsable:	Tannia Estefanny Duarte Vargas
Módulo:	Commits
Casos de prueba	
Precondiciones de la Prueba	<p>El usuario debe estar autenticado.</p> <p>Debe haber al menos un commit previo en el historial de la rama en la que se desea hacer el revert.</p> <p>El usuario debe tener permisos de modificación en la rama.</p> <p style="text-align: center;">Paso a Paso de la prueba:</p> <p>El usuario accede al historial de commits y selecciona el commit que desea revertir. El sistema muestra una opción para revertir el commit seleccionado.</p> <p>El usuario confirma la acción de revertir el commit.</p> <p>El sistema crea un nuevo commit que revierte los cambios realizados en el commit seleccionado.</p> <p>El sistema muestra una notificación de éxito indicando que el commit ha sido revertido.</p>
Postcondiciones de la prueba:	Un nuevo commit aparece en el historial, reflejando la reversión de los cambios realizados en el commit original.
Criterios de aceptación	<p>El sistema crea un commit de reversión exitosamente y se registra en el historial.</p> <p>Si el commit es el único en la rama o no se puede revertir, el sistema muestra un mensaje de error.</p>

Nota. Información de CP COM_CP12_Revertir_commit. *Fuente.* Autoría propia.

Tabla 20*Diseño de Prueba MER_CP13_Solicitud_merge*

Datos de la prueba	
Nombre de CP	MER_CP13_Solicitud_merge
Responsable:	Camilo Samudio
Módulo:	Merge
Casos de prueba	
Precondiciones de la Prueba	Tener acceso a tu repositorio en GitHub y de estar en la rama principal.
	Verifica que la rama que deseas fusionar.
	Paso a Paso de la prueba:
	Tener acceso a tu repositorio en GitHub y de estar en la rama principal con el comando <code>git checkout máster</code> .
	Verifica que la rama que deseas fusionar con el comando <code>git merge feature/new-feature</code> .
Postcondiciones de la prueba:	El código queda fusionado
Criterios de aceptación	Usar el comando <code>git merge feature/new-feature</code> de forma exitosa.

Nota. Información de CP MER_CP13_Solicitud_merge. *Fuente.* Autoría propia.

Tabla 21*Diseño de Prueba PRO_CP14_Crear_proyecto*

Datos de la prueba	
Nombre de CP	PRO_CP14_Crear_proyecto
Responsable:	Camilo Samudio
Módulo:	Projects
Casos de prueba	
Precondiciones de la Prueba	Tener acceso a tu cuenta en GitHub. Darle click a un nuevo proyecto.
Paso a Paso de la prueba:	
Darle click a un nuevo proyecto.	
Agregar nombre del proyecto y seleccionar el tipo de interfaz para el proyecto.	
Postcondiciones de la prueba:	El proyecto queda creado
Criterios de aceptación	Rellenar los datos necesarios para crear el proyecto.

Nota. Información de CP PRO_CP14_Crear_proyecto. *Fuente.* Autoría propia.

Tabla 22*Diseño de Prueba PRO_CP15_Cerrar_proyecto*

Datos de la prueba	
Nombre de CP	PRO_CP15_Cerrar_proyecto
Responsable:	Camilo Samudio
Módulo:	Projects
Casos de prueba	
Precondiciones de la Prueba	Tener acceso a tu cuenta en GitHub. Ingresar a la configuración del proyecto.
Paso a Paso de la prueba:	
Entra en la zona de peligro: En la página de configuración, busca la sección “Zona de peligro” y haz clic en el botón “cerrar project” (Close project).	
El proyecto estará cerrado: El proyecto será eliminado de la lista de proyectos y no se podrá acceder a él desde la página principal de GitHub.	
Postcondiciones de la prueba:	El proyecto queda cerrado
Criterios de aceptación	Darle click en Close project.

Nota. Información de CP PRO_CP15_Cerrar_proyecto. *Fuente.* Autoría propia.

Tabla 23*Diseño de Prueba PRO_CP16_Eliminar_proyecto*

Datos de la prueba	
Nombre de CP	PRO_CP16_Eliminar_proyecto
Responsable:	Camilo Samudio
Módulo:	Projects
Casos de prueba	
Precondiciones de la Prueba	Tener acceso a tu cuenta en GitHub. Ingresar a la configuración del proyecto.
Paso a Paso de la prueba:	
Entra en la zona de peligro: En la página de configuración, busca la sección “Zona de peligro” y haz clic en el botón “eliminar project” (Delete project).	
Confirma la eliminación: Lee las advertencias y escribe el nombre del proyecto en el cuadro de texto.	
El proyecto estará cerrado: El proyecto será eliminado de la lista de proyectos y no se podrá acceder a él desde la página principal de GitHub.	
Eliminar permanentemente un proyecto, debes tener permisos de administración en la organización o ser el propietario del repositorio.	
Postcondiciones de la prueba:	El proyecto queda eliminado
Criterios de aceptación	Darle click en Delete project.

Nota. Información de CP PRO_CP16_Eliminar_proyecto. *Fuente.* Autoría propia.

Tabla 24*Diseño de Prueba PRO_CP17_Adicion_items_proyecto*

Datos de la prueba	
Nombre de CP	PRO_CP17_Adicion_items_proyecto
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Projects
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un proyecto.
Paso a Paso de la prueba:	
El usuario ingresa al sistema GitHub con sus credenciales.	
El usuario selecciona el proyecto al que quiera agregar un ítem.	
El usuario selecciona la opción de “my items” y selecciona el tipo de ítem que desea crear, agrega el título, una descripción y hace clic en “crear”.	
El sistema crea y agrega el nuevo ítem al proyecto.	
Postcondiciones de la prueba:	El ítem se agrega exitosamente al proyecto.
Criterios de aceptación	El usuario debe ver el ítem creado en la lista de "my items" del proyecto. El sistema debe mostrar un mensaje de confirmación indicando que el ítem se ha creado correctamente. El ítem creado debe contener el título, la descripción y el tipo seleccionados por el usuario.

Nota. Información de CP PRO_CP17_Adicion_items_proyecto. *Fuente.* Autoría propia.

Tabla 25*Diseño de Prueba PRO_CP18_Adicion_miembros_proyecto*

Datos de la prueba	
Nombre de CP	PRO_CP18_Adicion_miembros_proyecto
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Repositories
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un repositorio al cual agregar miembros.
Paso a Paso de la prueba:	
	El usuario ingresa al sistema GitHub con sus credenciales.
	El usuario selecciona el repositorio al que quiera agregar un miembro.
	El usuario selecciona la opción de “configuración” y selecciona la sección “colaboradores”.
	El usuario hace clic en “agregar personas” e ingresa el nombre de usuario o correo de la persona que quiera agregar y hace clic en “agregar al repositorio”.
	El sistema agrega un nuevo miembro al proyecto.
Postcondiciones de la prueba:	El miembro se agrega exitosamente al proyecto.
Criterios de aceptación	El usuario debe ver el miembro agregado en la lista de miembros del proyecto. El sistema debe mostrar un mensaje de confirmación indicando que el ítem se ha creado correctamente. El miembro agregado debe los datos agregados por el usuario.

Nota. Información de CP PRO_CP18_Adicion_miembros_proyecto. *Fuente.* Autoría propia.

Tabla 26*Diseño de Prueba NOT_CP19_Recibir_notificaciones*

Datos de la prueba	
Nombre de CP	NOT_CP19_Recibir_notificaciones
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Notifications
Casos de prueba	
Precondiciones de la Prueba Haber iniciado sesión en la plataforma.	
Que el usuario interactúe con otros usuarios en la plataforma.	
Paso a Paso de la prueba:	
El usuario ingresa al sistema GitHub con sus credenciales.	
El sistema envía al usuario las notificaciones a partir de sus interacciones con otros usuarios.	
Postcondiciones de la prueba:	El usuario puede recibir correctamente sus notificaciones.
Criterios de aceptación	Recepción de Notificaciones. El sistema avisa al usuario al recibir las notificaciones.

Nota. Información de CP NOT_CP19_Recibir_notificaciones. *Fuente.* Autoría propia.

Tabla 27*Diseño de Prueba NOT_CP20_Leer_notificaciones*

Datos de la prueba	
Nombre de CP	NOT_CP20_Leer_notificaciones
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Notifications
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que el usuario interactúe con otros usuarios en la plataforma.
Paso a Paso de la prueba	
	El usuario ingresa al sistema GitHub con sus credenciales.
	El sistema envía al usuario las notificaciones a partir de sus interacciones con otros usuarios.
	El usuario selecciona el ícono de notificaciones y hace clic para ver las notificaciones recibidas.
Postcondiciones de la prueba:	El usuario puede ver y leer correctamente sus notificaciones.
Criterios de aceptación	El usuario puede ingresar al módulo de notificaciones. No existen notificaciones duplicadas. El usuario debe poder ver las notificaciones desde todos los dispositivos desde los que sea posible ingresar a la plataforma.

Nota. Información de CP NOT_CP20_Leer_notificaciones. *Fuente.* Autoría propia.

Tabla 28*Diseño de Prueba PAG_CP21_Alojar_pagina*

Datos de la prueba	
Nombre de CP	PAG_CP21_Alojar_pagina
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Pages
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un repositorio. Que el repositorio sea de una aplicación que se pueda alojar en GitHub Pages.
Paso a Paso de la prueba	
	El usuario ingresa al sistema GitHub con sus credenciales.
	El usuario crea un repositorio y almacena en este una aplicación que pueda ser alojada en GitHub Pages.
	El usuario selecciona el repositorio y hace clic en la opción de configuraciones.
	El usuario selecciona la opción “Pages” en configuraciones y selecciona la rama del repositorio que quiere que contiene la información que quiere que se aloje en GitHub Pages.
Postcondiciones de la prueba:	El usuario puede seleccionar la configuración de alojamiento en GitHub Pages.
Criterios de aceptación:	El sistema permite que el usuario seleccione la configuración correspondiente a la rama del proyecto que quiere alojar en GitHub Pages.

Nota. Información de CP PAG_CP21_Alojar_pagina. *Fuente.* Autoría propia.

Tabla 29*Diseño de Prueba PAG_CP22_Desplegar_pagina*

Datos de la prueba	
Nombre de CP	PAG_CP22_Desplegar_pagina
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Pages
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un repositorio alojado en GitHub Pages.
Paso a Paso de la prueba:	
	El usuario ingresa al sistema GitHub con sus credenciales.
	El selecciona el repositorio de la aplicación que quiere desplegar.
	El usuario confirma que la configuración de alojamiento esté completa y selecciona si quiere o no que la página tenga HTTPS.
	El usuario hace clic en “guardar” y espera que la página se despliegue.
Postcondiciones de la prueba:	El usuario puede acceder a la página desde el dominio que GitHub Pages le asigna.
Criterios de aceptación	GitHub despliega la página. El enlace asignado a la página por GitHub funciona.

Nota. Información de CP PAG_CP22_Desplegar_pagina. *Fuente.* Autoría propia

Tabla 30*Diseño de Prueba PAG_CP23_Personalizar_dominio*

Datos de la prueba	
Nombre de CP	PAG_CP23_Personalizar_dominio
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Pages
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un repositorio alojado y desplegado en GitHub Pages.
Paso a Paso de la prueba:	
El usuario ingresa al sistema GitHub con sus credenciales.	
El usuario selecciona el repositorio de la aplicación cuyo dominio en GitHub Pages quiere cambiar.	
El usuario cambia el dominio asignado por GitHub Pages a uno personalizado y guarda la configuración.	
GitHub Pages vuelve a desplegar la página.	
Postcondiciones de la prueba:	El usuario puede acceder a la página desde el dominio personalizado.
Criterios de aceptación	GitHub despliega la página correctamente con el dominio personalizado. El enlace personalizado funciona y el usuario puede acceder.

Nota. Información de CP PAG_CP23_Personalizar_dominio. *Fuente.* Autoría propia.

Tabla 31*Diseño de Prueba WIK_CP24_Crear_wiki*

Datos de la prueba	
Nombre de CP	WIK_CP24_Crear_wiki
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Wiki
Casos de prueba	
Precondiciones de la Prueba Haber iniciado sesión en la plataforma.	
Que exista un repositorio al que se le pueda crear una Wiki.	
Paso a Paso de la prueba:	
El usuario ingresa al sistema GitHub con sus credenciales.	
El selecciona el repositorio cuya Wiki quiera crear.	
El usuario selecciona la opción "Wiki" y hace clic en "Crear la primera Wiki" para agregar el título y el contenido de la Wiki.	
El usuario hace clic en "guardar página" para crear y guardar la Wiki.	
Postcondiciones de la prueba:	El usuario puede ver la Wiki que acaba de crear.
Criterios de aceptación	El sistema informa al usuario que se acaba de crear una nueva Wiki.
	El sistema muestra al usuario la Wiki que acaba de crear.

Nota. Información de CP WIK_CP24_Crear_wiki. *Fuente.* Autoría propia.

Tabla 32*Diseño de Prueba WIK_CP25_Editar_wiki*

Datos de la prueba	
Nombre de CP	WIK_CP25_Editar_wiki
Responsable:	Ichel Alejandra Delgado Morales
Módulo:	Wiki
Casos de prueba	
Precondiciones de la Prueba	Haber iniciado sesión en la plataforma. Que exista un repositorio con una Wiki creada.
Paso a Paso de la prueba	
El usuario ingresa al sistema GitHub con sus credenciales.	
El selecciona el repositorio cuya Wiki quiera editar.	
El usuario selecciona la opción “Wiki” y luego hace clic en “editar”.	
El usuario edita las opciones que desee editar en la Wiki.	
El usuario hace clic en “guardar página” para guardar los cambios que acaba de hacer.	
Postcondiciones de la prueba:	El usuario puede ver los cambios que acaba de hacer en la Wiki.
Criterios de aceptación	El sistema muestra al usuario los cambios que acaba de hacer en la Wiki.

Nota. Información de CP WIK_CP25_Editar_wiki. *Fuente.* Autoría propia

Ejecución de casos de prueba

Tabla 33

Ejecución de Caso de Prueba UP_CP01_Crear_cuenta

Nombre del caso de prueba: SUP_CP01_Crear_cuenta

Módulo: Sign up

Responsable: Camilo Samudio

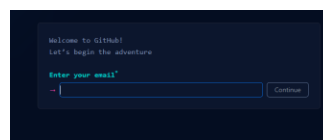
Precondiciones de la Prueba:

El usuario puede ingresar a la plataforma.

El usuario puede crear una cuenta.

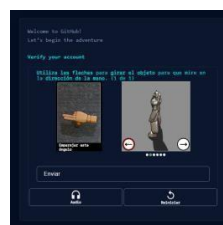
Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario selecciona la opción Sign up	El sistema debería mostrar una casilla para ingresar el email.	Figura 2 <i>Registro de correo</i>



Fuente. Autoría propia

El usuario introduce contraseña, Username y verifica que no sea un robot.	El sistema debe mostrar el formulario completo y pasar al captcha.	Figura 3 <i>Captcha</i>
---	--	-----------------------------------



Fuente. Autoría propia

Insertar código de verificación.	Interfaz muestra código de verificación.	Figura 4 Ingreso de código de verificación.
----------------------------------	--	---



Fuente. Autoría propia

Postcondiciones de la prueba:	Usuario ya está disponible para uso.
Criterios de aceptación	Exitoso (S/N)
Creación de usuario	S
Si el nombre del usuario ya existe, el sistema muestra un mensaje de error y no permite la creación, pero da la opción de un usuario similar.	S
Si la contraseña está en una lista de contraseñas comprometidas hay que cambiar a otra.	S
	Ejecutado:
	Exitoso <u>X</u>
Estado del caso de prueba	Fallido
	Detenido
	Pendiente de ejecución

Nota. Información de ejecución de UP_CP01_Crear_cuenta. *Fuente.* Autoría propia

Tabla 34*Ejecución Caso de Prueba LOG_CP02_Iniciar_sesion*

 Nombre del caso de prueba: LOG_CP02_Iniciar_sesion

Módulo: Log in

Responsable: Camilo Samudio

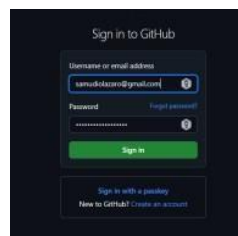
 Precondiciones de la Prueba:

El usuario puede ingresar a la plataforma

El usuario puede ingresar el username y password de forma correcta.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa el username y la contraseña	Interfaz con formulario para iniciar sesión	Figura 5 <i>Inicio sesión en plataforma</i>
Se le da al botón Sign in y se ingresa al dashboard de la aplicación.	Interfaz con las dashboard	Figura 6 <i>Interfaz de dashboard</i>
Postcondiciones de la prueba	El usuario está listo para utilizar Github	
Criterios de aceptación	Exitoso (S/N)	
Ingresar correctamente el usuario y la contraseña.	S	

*Fuente. Autoría propia**Fuente. Autoría propia*

	Ejecutado:
	Exitoso <u>X</u>
Estado del caso de prueba	Fallido
	Detenido
	Pendiente de ejecución

Nota. Información de ejecución de LOG_CP02_Iniciar_sesion. *Fuente.* Autoría propia

Tabla 35

Ejecución Caso de Prueba REP_CP03_Crear_repositorio


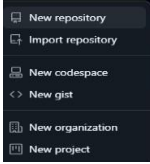

Nombre del caso de prueba: REP_CP03_Crear_repositorio

Módulo: Repositories

Responsable: Tannia Estefanny Duarte Vargas

Precondiciones de la Prueba:

El usuario debe haber iniciado sesión en el sistema y contar con permisos para crear repositorios.

Paso a Paso de la prueba		
Paso	Resultado	Soporte
El usuario selecciona la opción "Crear nuevo repositorio" desde el panel principal de repositorios.	El sistema debería mostrar un formulario para ingresar el nombre y la descripción del nuevo repositorio.	<p>Figura 7 <i>Inicio sesión en plataforma para crear repositorio</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 8 <i>Opción para crear nuevo repositorio</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 9 <i>Configuración para crear nuevo repositorio</i></p>  <p><i>Fuente. Autoría propia</i></p>

El usuario completa los campos del formulario, ingresando un nombre único y una descripción, y selecciona la opción "Crear".

El sistema verifica la unicidad del nombre del repositorio y realiza la creación en caso de ser único.

Figura 10

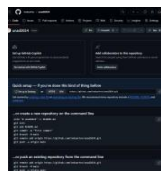
Verificación de datos de nuevo repositorio



Fuente. Autoría propia

Figura 11

Vista de repositorio



Fuente. Autoría propia

Una notificación confirma la creación exitosa del repositorio.

El nuevo repositorio aparece en la lista de repositorios del usuario en su panel de control.

Figura 12

Lista repositorios



Fuente. Autoría propia

Postcondiciones de la prueba: El usuario está listo para utilizar Github

Criterios de aceptación Exitoso (S/N)

Creación del repositorio sin errores S

Manejo de nombres duplicados S (Mensaje de error si es duplicado)

Estado del caso de prueba Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de REP_CP03_Crear_repositorio. Fuente. Autoría propia



Fuente. Autoría propia

Figura 17

Confirmación de eliminación de repositorio



Fuente. Autoría propia

Figura 18

Dashboard sin el repositorio



Fuente. Autoría propia

Confirmación de la eliminación por parte del usuario.

El sistema elimina el repositorio y muestra una notificación de éxito en pantalla.

Postcondiciones de la prueba

El repositorio debe desaparecer de la lista de repositorios del usuario autenticado.

Criterios de aceptación

Exitoso (S/N)

Eliminación efectiva del repositorio

S

Notificación de confirmación exitosa

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de REP_CP04_Eliminar_repositorio. Fuente. Autoría propia.

Tabla 37*Ejecución Caso de Prueba ISS_CP05_Crear_issue*

 Nombre del caso de prueba: ISS_CP05_Crear_issue

Módulo: Issues

Responsable: Tannia Estefanny Duarte Vargas

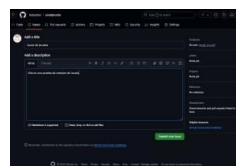
 Precondiciones de la prueba:

El usuario debe estar autenticado y tener permisos para crear issues en un repositorio específico.

Evidencia Visual: Captura de autenticación y permisos de usuario en el repositorio.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario selecciona la opción "Crear nuevo issue" dentro del repositorio.	El sistema debería mostrar un formulario para ingresar el título y la descripción del issue.	Figura 19 <i>Inicio sesión para Crear issue</i>

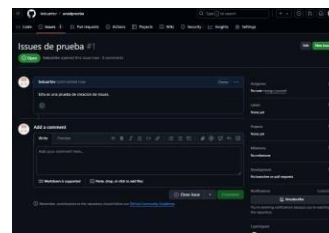
*Fuente. Autoría propia***Figura 20***Vista general de issues**Fuente. Autoría propia***Figura 21***Vista de opciones de creación de issue**Fuente. Autoría propia*

El usuario completa el formulario y selecciona "Crear".

El sistema crea el issue y muestra una notificación de creación exitosa; el issue aparece en la lista de issues abiertos.

Figura 22

Creación exitosa de issue



Fuente. Autoría propia

Postcondiciones de la prueba El nuevo issue debe estar visible en la lista de issues del repositorio.

Criterios de aceptación Exitoso (S/N)

Creación del issue sin errores S

Mensaje de error si falta el título o está duplicado S

Estado del caso de prueba

Ejecutado:
 Exitoso X
 Fallido
 Detenido
 Pendiente de ejecución

Nota. Información de ejecución de ISS_CP05_Crear_issue. Fuente. Autoría propia.

Tabla 38*Ejecución Caso de Prueba ISS_CP06_Cerrar_issue*

 Nombre del caso de prueba: ISS_CP06_Cerrar_issue

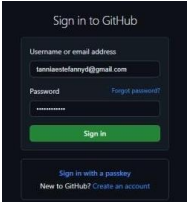

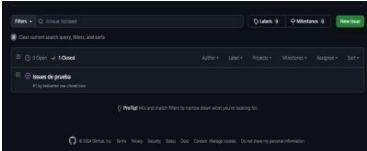
Módulo: Issues

Responsable: Tannia Estefanny Duarte Vargas

 Precondiciones de la prueba:

El usuario debe estar autenticado, tener permisos en el repositorio y debe haber al menos un issue abierto.

Evidencia Visual: Captura de autenticación y de la lista de issues abiertos.

Paso a Paso de la prueba		
Paso	Resultado	Soporte
Selección de un issue abierto y clic en la opción "Cerrar".	El sistema solicita una confirmación para cerrar el issue.	Figura 23 <i>Inicio sesión para cerrar issue</i> 
Confirmación del cierre por parte del usuario.	El sistema cambia el estado del issue a "Cerrado" y lo muestra en la lista de issues cerrados.	Figura 24 <i>Confirmación para cerrar issue</i> 
Postcondiciones de la prueba	El issue debe aparecer en la lista de issues cerrados del repositorio.	Figura 25 <i>Issue cerrado</i> 

Criterios de aceptación	Exitoso (S/N)
Cambio exitoso de estado a "Cerrado".	S
Mensaje de confirmación de cierre.	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de ISS_CP06_Cerrar_issue. *Fuente.* Autoría propia.

Tabla 39

Ejecución Caso de Prueba PR_CP07_Crear_pull_request

Nombre del caso de prueba: PR_CP07_Crear_pull_request

Módulo: Pull Requests

Responsable: Tannia Estefanny Duarte Vargas

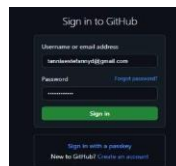
Precondiciones de la prueba:

El usuario debe tener cambios en una rama que no sea la principal y estar autenticado.

Evidencia Visual: Captura de autenticación y de los cambios en la rama.

Paso a Paso de la prueba

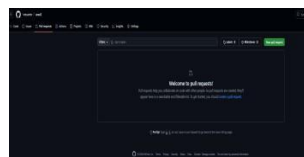
Paso	Resultado	Soporte
El usuario selecciona la opción "Crear nueva Pull Request".	El sistema muestra el formulario correspondiente.	Figura 26 <i>Inicio sesión para crear pull request</i>



Fuente. Autoría propia

Figura 27

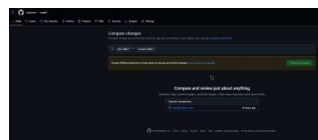
Vista general de los pull request del repositorio



Fuente. Autoría propia

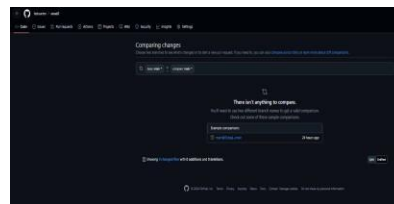
Figura 28

Vista de cambios



Fuente. Autoría propia

El usuario completa los campos y selecciona "Crear". La Pull Request se crea y aparece en la lista de Pull Requests pendientes.

Figura 29*Vista de pull request creado**Fuente. Autoría propia*

Postcondiciones de la prueba	La Pull Request debe estar visible y lista para ser revisada por otros usuarios con permisos.
Criterios de aceptación	Exitoso (S/N)
Creación exitosa de la Pull Request	S
Validación de permisos de usuario	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de PR_CP07_Crear_pull_request. *Fuente.* Autoría propia

Tabla 40*Ejecución Caso de Prueba PR_CP08_Revisar_pull_request*

 Nombre del caso de prueba: PR_CP08_Revisar_pull_request

Módulo: Pull Requests

Responsable: Tannia Estefanny Duarte Vargas

 Precondiciones de la prueba:

El usuario debe estar autenticado y tener permisos para revisar una Pull Request pendiente.

Evidencia Visual: Captura de autenticación y de la Pull Request pendiente.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Selección de la Pull Request para revisar.	El sistema muestra los cambios realizados en la Pull Request.	Figura 30 <i>Inicio sesión para revisar pull request</i>



Fuente. Autoría propia

Figura 31
Cambios realizados en el pull request

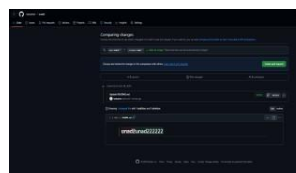


Fuente. Autoría propia

Selección de "Aprobar" o "Rechazar".

El estado de la Pull Request cambia de acuerdo con la revisión realizada.

Figura 32
Estado pull request



Fuente. Autoría propia

Postcondiciones de la prueba	La Pull Request debe reflejar el estado actualizado según la revisión efectuada.
Criterios de aceptación	Exitoso (S/N)
Estado actualizado de Pull Request	S
Notificación de cambio de estado	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido <u>Pendiente de ejecución</u>

Nota. Información de ejecución de PR_CP08_Revisar_pull_request. Fuente. Autoría propia.

Tabla 41*Ejecución Caso de Prueba BR_CP09_Crear_branch*

 Nombre del caso de prueba: BR_CP09_Crear_branch

Módulo: Branches



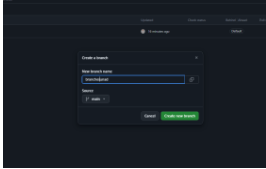
Responsable: Tannia Estefanny Duarte Vargas

 Precondiciones de la prueba:

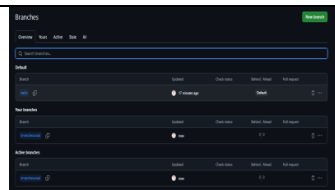
El usuario debe estar autenticado y tener permisos para crear ramas en el repositorio.

Evidencia Visual: Captura de autenticación y de la lista actual de ramas en el repositorio.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Selección de la opción "Crear nueva rama" desde la sección de branches del repositorio.	El sistema muestra un formulario para ingresar el nombre de la nueva rama y la opción para elegir la rama base.	Figura 33 <i>Inicio sesión para crear branch</i>  <p><i>Fuente. Autoría propia</i></p> Figura 34 <i>Vista general de ramas del repositorio</i>  <p><i>Fuente. Autoría propia</i></p> Figura 35 <i>Ventana de creación de rama nueva</i>  <p><i>Fuente. Autoría propia</i></p>
Ingreso de nombre único para la rama y selección de	El sistema verifica la unicidad del nombre y	Figura 36 <i>Rama nueva creada</i>

"Crear".
crea la nueva rama si el nombre es único.



Fuente. Autoría propia

Postcondiciones de la prueba	La nueva rama debe estar visible en la lista de ramas del repositorio.
Criterios de aceptación	Exitoso (S/N)
Creación de rama sin errores	S
Verificación de unicidad de Nombre	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de BR_CP09_Crear_branch. *Fuente.* Autoría propia.

Tabla 42*Ejecución Caso de Prueba BR_CP10_Eliminar_branch*

 Nombre del caso de prueba: BR_CP10_Eliminar_branch

Módulo: Branches


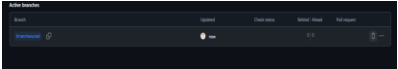
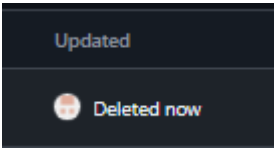
Responsable: Tannia Estefanny Duarte Vargas

 Precondiciones de la prueba:

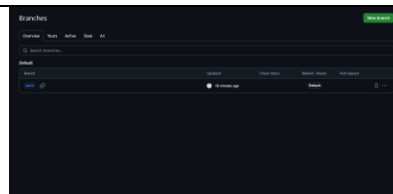
El usuario debe estar autenticado y tener permisos de eliminación en el repositorio, con al menos una rama que no sea la rama principal.

Evidencia Visual: Captura de autenticación y de la lista de ramas, indicando cuál se va a eliminar.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Selección de una rama existente que el usuario desea eliminar y clic en la opción "Eliminar".	El sistema muestra una advertencia de confirmación de eliminación.	Figura 37 <i>Inicio sesión para eliminar branch</i>  <p>Fuente. Autoría propia</p> Figura 38 <i>Visualización de listado de branches</i>  <p>Fuente. Autoría propia</p> Figura 39 <i>Opción de eliminar Branch</i>  <p>Fuente. Autoría propia</p>
Confirmación de eliminación por el usuario.	El sistema elimina la rama seleccionada y	Figura 40 <i>Eliminar rama</i>

notifica el éxito de la acción.



Fuente. Autoría propia

Postcondiciones de la prueba:	La rama eliminada debe desaparecer de la lista de ramas del repositorio.
Criterios de aceptación	Exitoso (S/N)
Eliminación de la rama sin errores	S
Confirmación de acción de eliminación	S

Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución
---------------------------	---

Nota. Información de ejecución de BR_CP10_Eliminar_branch. *Fuente.* Autoría propia.

Tabla 43

Ejecución Caso de Prueba COM_CP11_Hacer_commit

Nombre del caso de prueba: COM_CP11_Hacer_commit

Módulo: Commits

Responsable: Tannia Estefanny Duarte Vargas

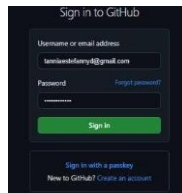
Precondiciones de la prueba:

El usuario debe estar autenticado y tener permisos para realizar commits en una rama específica.

Evidencia Visual: Captura de autenticación y del repositorio con cambios pendientes por realizar.

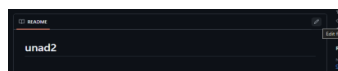
Paso a Paso de la prueba

Paso	Resultado	Soporte
Selección de los cambios no guardados en la rama y clic en "Hacer commit".	El sistema muestra un formulario para ingresar el mensaje del commit.	Figura 41 <i>Inicio sesión para realizar commit</i>



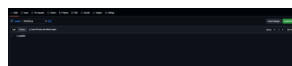
Fuente. Autoría propia

Figura 42
Visualización de repositorio para commit



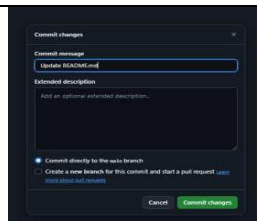
Fuente. Autoría propia

Figura 43
Creación de commit dentro de la rama unad2



Fuente. Autoría propia

Figura 44
Confirmación de creación de commit



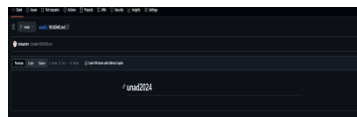
Fuente. Autoría propia

Ingreso de un mensaje descriptivo y clic en "Confirmar".

El sistema guarda los cambios en el repositorio y actualizar el historial de commits de la rama.

Figura 45

Actualización de historial de commits



Fuente. Autoría propia

Postcondiciones de la prueba: El commit debe estar visible en el historial de la rama seleccionada.

Criterios de aceptación

Exitoso (S/N)

Commit sin errores en el sistema

S

Notificación de éxito del commit

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de COM_CP11_Hacer_commit. Fuente. Autoría propia.

Tabla 44*Ejecución Caso de Prueba de Software COM_CP12_Revertir_commit*

 Nombre del caso de prueba: COM_CP12_Revertir_commit

Módulo: Commits


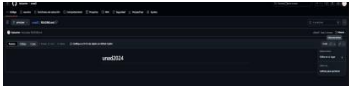

Responsable: Tannia Estefanny Duarte Vargas

 Precondiciones de la prueba:

El usuario debe estar autenticado y tener permisos de edición en la rama donde se desea revertir un commit.

Evidencia Visual: Captura de autenticación y de un historial de commits con el commit a revertir.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Selección del commit específico que el usuario desea revertir y clic en la opción "Revertir".	El sistema muestra una advertencia de confirmación para la reversión del commit.	Figura 46 <i>Inicio de sesión para revertir commit</i>  <p><i>Fuente. Autoría propia</i></p> Figura 47 <i>Vista general de repositorio con commit a revertir</i>  <p><i>Fuente. Autoría propia</i></p> Figura 48 <i>Advertencia de reversión de commit</i>  <p><i>Fuente. Autoría propia</i></p>
Confirmación de reversión del	El sistema crea un	Figura 49

commit.

nuevo commit que
reverte los cambios
del commit original,
actualizando el
historial de commits.

Crea nuevo commit de reversión



Fuente. Autoría propia

Figura 50

Vista de commit de reversión



Fuente. Autoría propia

Postcondiciones de la prueba:	El historial de commits debe reflejar el nuevo commit de reversión y los cambios originales revertidos.
Criterios de aceptación	Exitoso (S/N)
Creación de commit de reversión sin errores	S
Notificación de éxito de reversión	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de COM_CP12_Revertir_commit. Fuente. Autoría propia.

Tabla 45*Ejecución Caso de Prueba MER_CP13_Solicitud_merge*

 Nombre del caso de prueba: MER_CP13_Solicitud_merge

Módulo: Log in

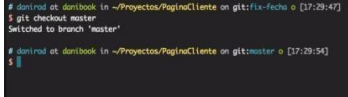

Responsable: Camilo Samudio

 Precondiciones de la prueba:

Tener acceso a tu repositorio en GitHub y de estar en la rama principal.

Verifica que la rama que deseas fusionar.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Tener acceso a tu repositorio en GitHub y de estar en la rama principal con el comando git checkout máster.	Muestra la rama de destino	Figura 51 <i>Rama destino</i>  <p>Fuente. Autoría propia</p>
Verifica que la rama que deseas fusionar con el comando git merge feature/new- feature.	Fusiona las ramas seleccionadas	Figura 52 <i>Rama fusionada</i>  <p>Fuente. Autoría propia</p>
Postcondiciones de la prueba		El código queda fusionado.
Criterios de aceptación		Exitoso (S/N)
Usar el comando git merge feature/new- feature de forma exitosa.		S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución	

Nota. Información de ejecución de MER_CP13_Solicitud_merge. Fuente. Autoría propia.

Tabla 46*Ejecución Caso de Prueba PRO_CP14_Crear_proyecto*

 Nombre del caso de prueba: PRO_CP14_Crear_proyecto

Módulo: Projects

Responsable: Camilo Samudio

 Precondiciones de la prueba:

Tener acceso a tu cuenta en GitHub.

Darle click a un nuevo proyecto.

Paso a Paso de la prueba



Paso	Resultado	Soporte
Darle click a un nuevo proyecto.	Interfaz para crear nuevo proyecto	Figura 53 <i>Crear proyecto</i>  <i>Fuente. Autoría propia</i>
Agregar nombre del proyecto	Nuevo proyecto creado	Figura 54 <i>Proyecto creado</i>  <i>Fuente. Autoría propia</i>
Postcondiciones de la prueba	El proyecto queda creado	
Criterios de aceptación	Exitoso (S/N)	
Rellenar los datos necesarios para crear el proyecto	S	
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución	

Tabla 47*Ejecución Caso de Prueba PRO_CP15_Cerrar_project*

Nombre del caso de prueba: PRO_CP15_Cerrar_projecto

Módulo: Projects

Responsable: Camilo Samudio

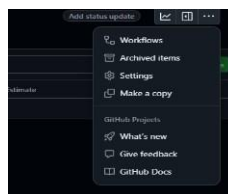
Precondiciones de la prueba:

Tener acceso a tu cuenta en GitHub.

Darle click a cerrar proyecto.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Darle click a configuraciones.	Menú de configuración	Figura 55 <i>Menú configuración cerrar proyecto</i>
Ir a la zona de peligro	Interfaz con zona de peligro	<i>Fuente. Autoría propia</i> Figura 56 <i>Zona de peligro cerrar proyecto</i>
Dar click en cerrar proyecto	Interfaz con zona de peligro	<i>Fuente. Autoría propia</i> Figura 57 <i>Cerrar proyecto</i>
Postcondiciones de la prueba	El proyecto queda cerrado para todos los usuarios	
Criterios de aceptación	Exitoso (S/N)	
Darle click en Close project.	S	
Estado del caso de prueba	Ejecutado:	



Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de PRO_CP15_Cerrar_proyecto. *Fuente.* Autoría propia

Tabla 48

Ejecución Caso de Prueba PRO_CP16_Eliminar_proyecto

Nombre del caso de prueba: PRO_CP16_Eliminar_proyecto

Módulo: Projects

Responsable: Camilo Samudio

Precondiciones de la prueba:

Tener acceso a tu cuenta en GitHub.

Darle click a eliminar proyecto.

Paso a Paso de la prueba

Paso	Resultado	Soporte
Darle click a configuraciones.	Menú de configuración	<p>Figura 58 <i>Menú de configuración eliminar proyecto</i></p>  <p><i>Fuente. Autoría propia</i></p>
Ir a la zona de peligro	Interfaz con zona de peligro	<p>Figura 59 <i>Zona de peligro eliminar proyecto</i></p>  <p><i>Fuente. Autoría propia</i></p>
Dar click en eliminar proyecto	Interfaz con zona de peligro	<p>Figura 60 <i>Eliminar proyecto</i></p>  <p><i>Fuente. Autoría propia</i></p>
Postcondiciones de la prueba	El proyecto queda cerrado para todos los usuarios	

Criterios de aceptación	Exitoso (S/N)
Darle click en eliminar project.	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de PRO_CP16_Eliminar_proyecto. *Fuente.* Autoría propia

Tabla 49

Ejecución Caso de Prueba PRO_CP17_Adicion_items_proyecto

Nombre del caso de prueba: PRO_CP17_Adicion_items_proyecto

Módulo: Projects

Responsable: Ichel Alejandra Delgado Morales

Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que el usuario interactúe con otros usuarios en la plataforma.

Paso a Paso de la prueba		
Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	<p>Figura 61</p> <p><i>Inicio de sesión para adicionar proyectos</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 62</p> <p><i>Página inicial de repositorios de usuario</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 63</p> <p><i>Vista general de proyectos</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 64</p> <p><i>Vista general de proyecto seleccionado</i></p>
El usuario selecciona el proyecto al que quiera agregar un ítem.	Se selecciona el ítem y la plataforma redirige al usuario a la página del proyecto.	<p>Figura 63</p> <p><i>Vista general de proyectos</i></p>  <p><i>Fuente. Autoría propia</i></p> <p>Figura 64</p> <p><i>Vista general de proyecto seleccionado</i></p>



Fuente. Autoría propia

El usuario selecciona la opción de “my items” y selecciona el tipo de ítem que desea crear, agrega el título, una descripción y hace clic en “crear”.

El usuario puede agregar los datos requeridos para crear el ítem nuevo.

Figura 65

Agregar datos nuevo ítem



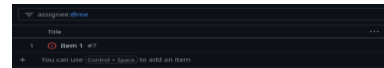
Fuente. Autoría propia

El sistema crea y agrega el nuevo ítem al proyecto.

Se puede ver el ítem creado y agregado al listado de ítems.

Figura 66

Item creado



Fuente. Autoría propia

Postcondiciones de la prueba

El ítem se agrega exitosamente al proyecto.

Criterios de aceptación

Exitoso (S/N)

El usuario debe ver el ítem creado en la lista de "my items" del proyecto. S

El sistema debe mostrar un mensaje de confirmación indicando que el ítem se ha creado correctamente. S

El ítem creado debe contener el título, la descripción y el tipo seleccionados por el usuario. S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de PRO_CP17_Adicion_items_proyecto. Fuente. Autoría propia

Tabla 50*Ejecución Caso de Prueba PRO_CP18_Adicion_miembros_proyecto*

 Nombre del caso de prueba: PRO_CP18_Adicion_miembros_proyecto

Módulo: Repositories

Responsable: Ichel Alejandra Delgado Morales

 Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que exista un repositorio al cual agregar miembros.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 67 <i>Inicio sesión para adición de miembros</i>

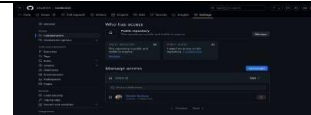
*Fuente. Autoría propia***Figura 68***Página inicial adición miembros**Fuente. Autoría propia***Figura 69***Vista general de página repositorio**Fuente. Autoría propia*

El usuario selecciona el repositorio al que quiera agregar un miembro.	Se selecciona la plataforma redirige al usuario a la página del repositorio.
--	--

El usuario selecciona la opción de “configuración” y selecciona la sección	El usuario puede agregar los datos requeridos para crear el ítem nuevo.
--	---

Figura 70*Adición de datos de nuevo miembro*

“colaboradores”.



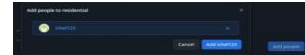
Fuente. Autoría propia

El usuario hace clic en “agregar personas” e ingresa el nombre de usuario o correo de la persona que quiera agregar y hace clic en “agregar al repositorio”.

Se puede ver el nombre de usuario o correo del miembro a agregar y el botón que dice “agregar <miembro>”.

Figura 71

Adición de nuevo miembro



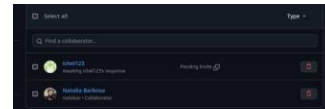
Fuente. Autoría propia

El sistema agrega un nuevo miembro al proyecto.

Se puede ver el miembro agregado al listado de colaboradores.

Figura 72

Lista de miembros de proyectos



Fuente. Autoría propia

Postcondiciones de la prueba:

El colaborador se agrega exitosamente al proyecto.

Criterios de aceptación

Exitoso (S/N)

El usuario debe ver el ítem creado en la lista de "my items" del proyecto.

S

El sistema debe mostrar un mensaje de confirmación indicando que el ítem se ha creado correctamente.

S

El ítem creado debe contener el título, la descripción y el tipo seleccionados por el usuario.

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de PRO_CP18_Adicion_miembros_proyecto. Fuente. Autoría propia

Tabla 51

Ejecución Caso de Prueba NOT_CP19_Recibir_notificaciones

Nombre del caso de prueba: NOT_CP19_Recibir_notificaciones

Módulo: Notifications

Responsable: Ichel Alejandra Delgado Morales

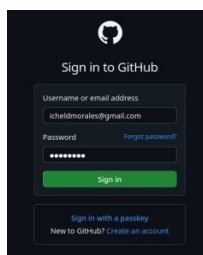
Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que el usuario interactúe con otros usuarios en la plataforma.

Paso a Paso de la prueba

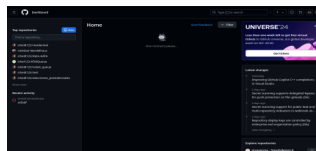
Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 73 <i>Iniciar sesión para recibir notificaciones</i>



Fuente. Autoría propia

Figura 74

Plataforma inicial para recibir notificaciones



Fuente. Autoría propia

El sistema envía al usuario las notificaciones a partir de sus interacciones con otros usuarios.	Se puede ver el ícono de notificaciones con un punto azul, indicando que se recibieron notificaciones.	Figura 75 <i>Icono notificaciones</i>
--	--	---



Fuente. Autoría propia

Postcondiciones de la prueba	El usuario puede recibir correctamente sus notificaciones.
Criterios de aceptación	Exitoso (S/N)
Recepción de Notificaciones.	S
El sistema avisa al usuario al recibir las notificaciones.	S
Estado del caso de prueba	Ejecutado: Exitoso <u>X</u> Fallido Detenido Pendiente de ejecución

Nota. Información de ejecución de NOT_CP19_Recibir_notificaciones. *Fuente.* Autoría propia

Tabla 52*Ejecución Caso de Prueba NOT_CP20_Leer_notificaciones*

Nombre del caso de prueba: NOT_CP20_Leer_notificaciones

Módulo: Notifications

Responsable: Ichel Alejandra Delgado Morales

Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que el usuario interactúe con otros usuarios en la plataforma.

Paso a Paso de la prueba

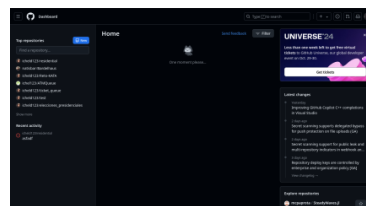
Paso

Resultado

Soporte

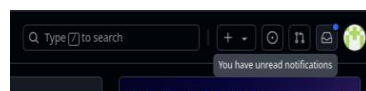
El usuario ingresa al sistema GitHub con sus credenciales.

Se ingresa a la plataforma y se ve la página inicial.

Figura 76*Iniciar sesión para leer notificaciones**Fuente. Autoría propia***Figura 77***Página inicial para leer notificaciones**Fuente. Autoría propia***Figura 78***Icono notificación para leer notificaciones*

El sistema envía al usuario las notificaciones a partir de sus interacciones con otros usuarios.

Se puede ver el ícono de notificaciones con un punto azul, indicando que se recibieron notificaciones.

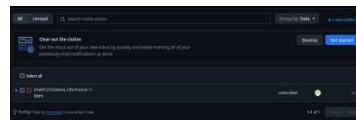
*Fuente. Autoría propia*

El sistema envía al usuario las notificaciones a partir de dichas interacciones.

Se puede acceder a las notificaciones y ver el detalle de cada una.

Figura 79

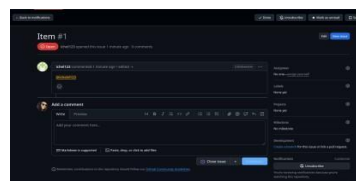
Visualización de las notificaciones



Fuente. Autoría propia

Figura 80

Visualización de detalle notificación



Fuente. Autoría propia

Postcondiciones de la prueba El usuario puede ver y leer correctamente sus notificaciones.

Criterios de aceptación

Exitoso (S/N)

El usuario puede ingresar al módulo de notificaciones.

S

No existen notificaciones duplicadas.

S

El usuario debe poder ver las notificaciones desde todos los dispositivos desde los que sea posible ingresar a la plataforma.

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de NOT_CP20_Leer_notificaciones. Fuente. Autoría propia

Tabla 53*Ejecución Caso de Prueba PAG_CP21_Alojar_pagina*

 Nombre del caso de prueba: PAG_CP21_Alojar_pagina

Módulo: Pages

Responsable: Ichel Alejandra Delgado Morales




 Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que exista un repositorio.

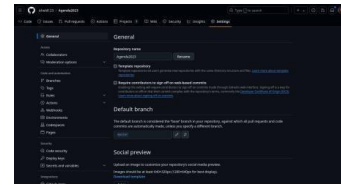
Que el repositorio sea de una aplicación que se pueda alojar en GitHub Pages.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 81 <i>Iniciar sesión para alojar página</i> 
		<i>Fuente. Autoría propia</i> Figura 82 <i>Página inicial de repositorios para alojar</i> 
El usuario crea un repositorio y almacena en este una aplicación que pueda ser alojada en GitHub Pages.	Visualización de la aplicación o proyecto almacenado en el repositorio.	Figura 83 <i>Visualización de proyectos para Alojar</i> 
El usuario selecciona el	Se puede ingresar a la	<i>Fuente. Autoría propia</i> Figura 84

repositorio y hace clic en la configuración del repositorio. opción de configuraciones.

Configuración repositorios de alojamiento



Fuente. Autoría propia

El usuario selecciona la opción “Pages” en configuraciones y selecciona la rama del repositorio que quiere que contenga la información que quiere que se aloje en GitHub Pages.

Se puede acceder a GitHub Pages y configurar los datos necesarios para guardar el alojamiento del proyecto.

Figura 85
Guardar alojamiento del proyecto



Fuente. Autoría propia

Postcondiciones de la prueba

El usuario puede guardar la configuración de alojamiento en GitHub Pages.

Criterios de aceptación

Exitoso (S/N)

El sistema permite que el usuario seleccione la configuración correspondiente a la rama del proyecto que quiere alojar en GitHub Pages y asigna un dominio.

S

Estado del caso de prueba

- Ejecutado:
- Exitoso X
- Fallido
- Detenido
- Pendiente de ejecución

Nota. Información de ejecución de PAG_CP21_Alojar_pagina. Fuente. Autoría propia

Tabla 54*Ejecución Caso de Prueba PAG_CP22_Desplegar_pagina*

 Nombre del caso de prueba: PAG_CP22_Desplegar_pagina

Módulo: Pages

Responsable: Ichel Alejandra Delgado Morales

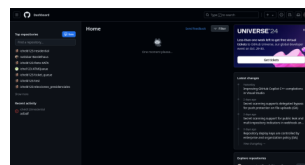
 Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

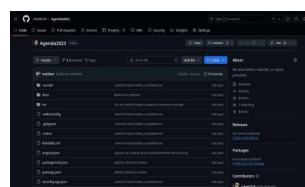
Que exista un repositorio alojado en GitHub Pages.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 86 <i>Iniciar sesión para desplegar proyecto</i>

*Fuente. Autoría propia***Figura 87***Página inicial para desplegar proyecto**Fuente. Autoría propia*

El usuario selecciona el repositorio de la aplicación que quiere desplegar.	Visualización de la aplicación o proyecto almacenado en el repositorio y que se quiere desplegar.	Figura 88 <i>Visualización de proyectos a desplegar</i>
---	---	---

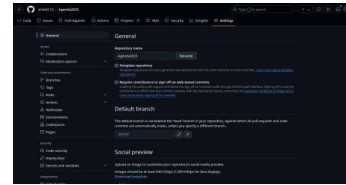
*Fuente. Autoría propia*

El usuario confirma que la configuración de alojamiento esté completa y selecciona si quiere o no que la página tenga HTTPS.

Se puede acceder a GitHub Pages y confirmar la configuración.

Figura 89

Configuración de despliegue de proyecto



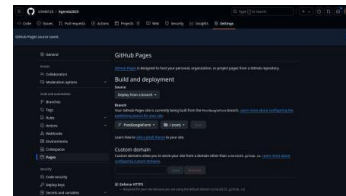
Fuente. Autoría propia

El usuario hace clic en “guardar” y espera que la página se despliegue.

El sistema guarda la configuración creada por el usuario y muestra el dominio asignado para su página.

Figura 90

Guarda configuración de despliegue



Fuente. Autoría propia

Figura 91

Configuración final de despliegue



Fuente. Autoría propia

Postcondiciones de la prueba

El usuario puede acceder a la página desde el dominio que GitHub Pages le asigna.

Criterios de aceptación

Exitoso (S/N)

GitHub despliega la página.

S

El enlace asignado a la página por GitHub funciona.

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de PAG_CP22_Desplegar_pagina. Fuente. Autoría propia

Tabla 55*Ejecución Caso de Prueba PAG_CP23_Personalizar_dominio*

 Nombre del caso de prueba: PAG_CP23_Personalizar_dominio

Módulo: Pages




Responsable: Ichel Alejandra Delgado Morales

 Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que exista un repositorio alojado y desplegado en GitHub Pages.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 92 <i>Iniciar sesión para personalizar dominio</i>
		
		<i>Fuente. Autoría propia</i> Figura 93 <i>Página inicial para personalizar dominio</i>
		
		<i>Fuente. Autoría propia</i> Figura 94 <i>Visualización de proyectos para Personalizar</i>
El usuario selecciona el repositorio de la aplicación cuyo dominio en GitHub Pages quiere cambiar.	Visualización de la aplicación o proyecto cuyo dominio se quiere personalizar.	

Fuente. Autoría propia

El usuario cambia el dominio asignado por GitHub Pages a uno personalizado y guarda la configuración. Se puede cambiar el dominio, se valida el DNS y se guarda la configuración.

Figura 95*Cambio dominio de proyecto**Fuente. Autoría propia***Figura 96***Guarda configuración de dominio**Fuente. Autoría propia*

GitHub Pages vuelve a desplegar la página.

El sistema despliega nuevamente la página con el nuevo dominio.

Figura 97*Página con nuevo dominio**Fuente. Autoría propia*

Postcondiciones de la prueba

El usuario puede acceder a la página desde el dominio personalizado.

Criterios de aceptación

Exitoso (S/N)

GitHub despliega la página correctamente con el dominio personalizado.

S

El enlace personalizado funciona y el usuario puede acceder. S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de PAG_CP23_Personalizar_dominio. Fuente. Autoría propia

Tabla 56

Ejecución Caso de Prueba WIK_CP24_Crear_wiki

Nombre del caso de prueba: WIK_CP24_Crear_wiki

Módulo: Wiki

Responsable: Ichel Alejandra Delgado Morales

Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

Que exista un repositorio al que se le pueda crear una Wiki.

Paso a Paso de la prueba

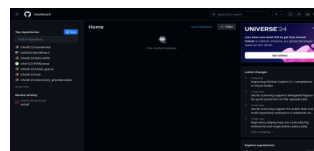
Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 98 <i>Iniciar sesión para crear wiki</i>



Fuente. Autoría propia

Figura 99

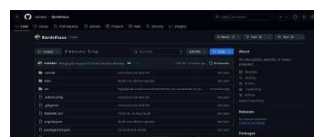
Página inicial de proyectos para crear wiki



Fuente. Autoría propia

Figura 100

El selecciona el repositorio cuya Wiki quiera crear.	Visualización de la aplicación o proyecto cuya Wiki quiere crear.	<i>Visualización de proyectos para crear wiki</i>
--	---	---



Fuente. Autoría propia

El usuario selecciona la opción “Wiki” y hace clic en “Crear primera Wiki” para agregar el título y el contenido de la Wiki.

Se puede acceder al módulo de Wiki y crear la primera página de esta.

Figura 101

Visualización de sección de wiki



Fuente. Autoría propia

El usuario hace clic en “guardar página” para crear y guardar la Wiki.

Se puede guardar la creación de la primera página y visualizarla.

Figura 102

Creación de página de wiki



Fuente. Autoría propia

Figura 103

Visualización de página de wiki creada



Fuente. Autoría propia

Postcondiciones de la prueba

El usuario puede ver la Wiki que acaba de crear.

Criterios de aceptación

Exitoso (S/N)

El sistema informa al usuario que se acaba de crear una nueva Wiki.

S

El sistema muestra al usuario la Wiki que acaba de crear.

S

Estado del caso de prueba

Ejecutado:

Exitoso X

Fallido

Detenido

Pendiente de ejecución

Nota. Información de ejecución de WIK_CP24_Crear_wiki. Fuente. Autoría propia

Tabla 57*Ejecución Caso de Prueba WIK_CP25_Editar_wiki*

 Nombre del caso de prueba: WIK_CP25_Editar_wiki

Módulo: Wiki

Responsable: Ichel Alejandra Delgado Morales

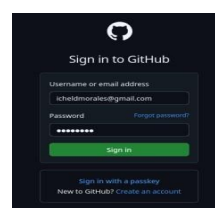
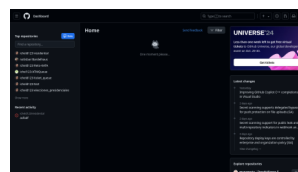
 Precondiciones de la prueba:

Haber iniciado sesión en la plataforma.

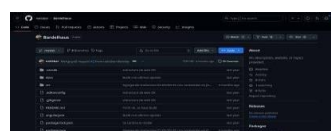
Que exista un repositorio con una Wiki creada.

Paso a Paso de la prueba

Paso	Resultado	Soporte
El usuario ingresa al sistema GitHub con sus credenciales.	Se ingresa a la plataforma y se ve la página inicial.	Figura 104 <i>Iniciar sesión para editar wiki</i>

*Fuente. Autoría propia***Figura 105***Página inicial de proyectos para editar wiki**Fuente. Autoría propia*

El selecciona el repositorio cuya Wiki quiera editar.	Visualización de la aplicación o proyecto cuya Wiki quiere editar.	Figura 106 <i>Visualización de proyectos para editar wiki</i>
---	--	---

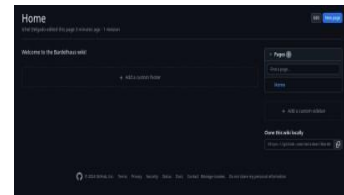
*Fuente. Autoría propia*

El usuario selecciona la opción	Se puede acceder al	Figura 107
---------------------------------	---------------------	-------------------

“Wiki” y luego hace clic en “editar”.

módulo de Wiki ver la ventana de edición de esta.

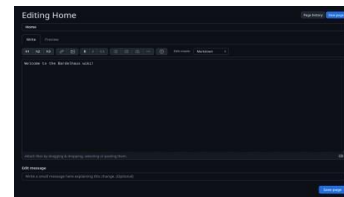
Visualización de módulo wiki



Fuente. Autoría propia

Figura 108

Ventana de edición de wiki



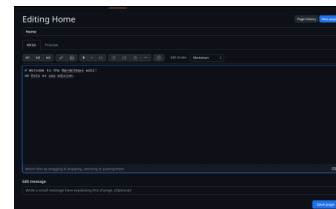
Fuente. Autoría propia

Figura 109

Edición de opciones de wiki

El usuario edita las opciones que desee editar en la Wiki.

Se pueden editar las opciones de la página de la Wiki seleccionada.



Fuente. Autoría propia

Figura 110

Confirmación de edición de wiki

El usuario hace clic en “guardar página” para guardar los cambios que acaba de hacer.

Se puede guardar la edición de la primera página y visualizarla.



Fuente. Autoría propia

Postcondiciones de la prueba

El usuario puede ver los cambios que acaba de hacer en la Wiki.

Criterios de aceptación

Exitoso (S/N)

El sistema muestra al usuario

los cambios que acaba de hacer en la Wiki. S

Estado del caso de prueba	Ejecutado:
	Exitoso <u>X</u>
	Fallido
	Detenido
	Pendiente de ejecución

Nota. Información de ejecución de WIK_CP25_Editar_wik. *Fuente.* Autoría propia

Gestión de defectos

La gestión de defectos es un proceso fundamental y estratégico en el desarrollo de software y servicios tecnológicos. Este proceso implica una serie de actividades esenciales que abarcan la identificación, registro, clasificación, evaluación y resolución de errores, fallas o carencias detectadas en los productos, tanto en el software como en la documentación asociada, durante su fase de desarrollo.

Identificar defectos en etapas tempranas permite mitigar riesgos y reducir costos asociados a la corrección de errores en fases posteriores. El registro adecuado de cada defecto garantiza un seguimiento eficiente, proporcionando información detallada que incluye el origen, la descripción, el impacto y la prioridad de cada incidencia.

La clasificación de defectos es crucial para determinar la gravedad e impacto en el sistema, facilitando la toma de decisiones en cuanto a los recursos y tiempo requeridos para su resolución. La evaluación, por su parte, permite priorizar los defectos según su relevancia y urgencia, asegurando que se resuelvan primero aquellos que puedan comprometer la funcionalidad crítica del sistema.

En este proyecto, los defectos identificados durante la ejecución de las pruebas se documentan y organizan de manera meticulosa en la tabla 4 “Gestión de incidencias”, lo que facilita su análisis y seguimiento continuo:

Tabla 58*Gestión de incidencias*

#	Fecha	Responsable	Módulo	Caso de Prueba	Descripción Defecto	Naturaleza	Tipo Incidencia	Severidad	Prioridad
1	27/10/2024	Ichel Alejandra Delgado Morales	Notifications	NOT_CP19_Recibir notificaciones	El sistema, por defecto, sólo envía notificaciones al correo, por lo que se debe habilitar la opción de recibir notificaciones a través de la plataforma para poder recibirlas.	Funcionalidad	Sugerencia	Medio	Normal
2	28/10/2024	Tannia Estefanny Duarte Vargas	Repositorios / Crear Issue	ISS_CP05_Crear_issue	No se permite crear un issue sin título.	Funcional	Error	Alta	Urgente
3	28/10/2024	Tannia Estefanny Duarte Vargas	Repositorios / Eliminar Repo	REP_CP07_Eliminar_repositorio	Error al eliminar o repositorios.	Funcional	Bug	Media	Alta
4	28/10/2024	Tannia Estefanny Duarte Vargas	Branches / Eliminar Branch	BR_CP10_Eliminar_branch	No se actualiza la lista tras eliminación.	Interfaz de Usuario	Error	Baja	Media

5	28/10/2024	Camilo Samudio	Projects	PRO_CP14_Crear_proyecto	No vuelve a la lista de proyectos tras crear el nuevo proyecto.	Funcionalidad	sugerencia	Baja	Baja
---	------------	----------------	----------	-------------------------	---	---------------	------------	------	------

Nota. Información obtenida a partir de la gestión de incidencia. *Fuente.* Autoría propia

Métricas

Las métricas de pruebas de software son herramientas fundamentales para la evaluación y control de la calidad del software en desarrollo. Estas métricas proporcionan datos cuantitativos que permiten medir la efectividad y eficiencia del proceso de pruebas, así como el nivel de calidad del producto final. Según Sommerville (2011), las métricas de pruebas no solo ayudan a gestionar el progreso del proyecto, sino también a identificar áreas de riesgo y a tomar decisiones informadas sobre las siguientes etapas del desarrollo.

Métricas de Ejecución

En la siguiente tabla se muestra el porcentaje de casos de pruebas exitosos y no exitosos por cada módulo:

Tabla 59

Métricas de ejecución

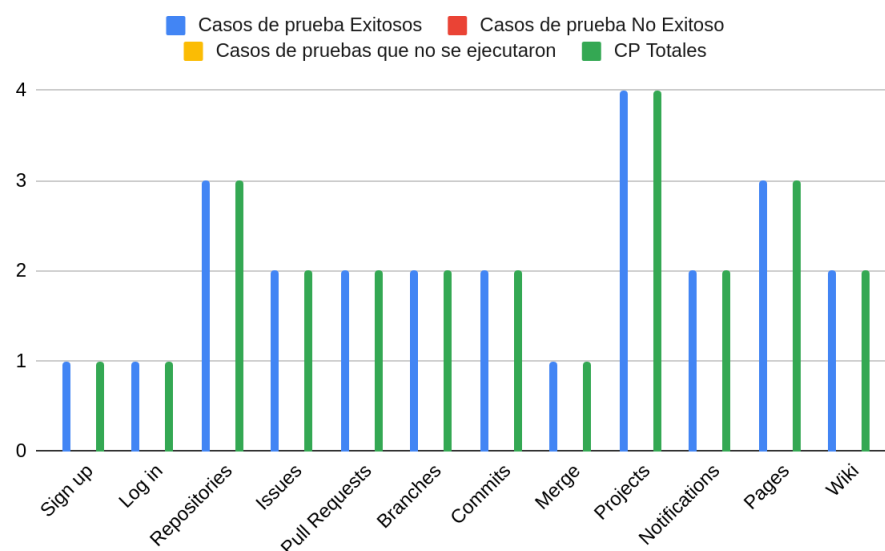
Responsable	Módulo	Casos de prueba Exitosos		Casos de prueba No Exitoso		Casos de pruebas que no se ejecutaron		CP Totales
		#	%	#	%	#	%	
Camilo Andres Samudio Lazaro	Sign up	1	1/1	0	0/1	0	0/1	1
Ichel Alejandra Delgado Morales / Tannia	Log in	1	1/1	0	0/1	0	0/1	1
Estefanny Duarte Vargas	Repositories	3	3/3	0	0/3	0	0/3	3
Tannia	Pull Requests	2	2/2	0	0/2	0	0/2	2
Estefanny Duarte Vargas	Branches	2	2/2	0	0/2	0	0/2	2
Camilo Andres Samudio Lazaro	Commits	2	2/2	0	0/2	0	0/2	2
	Merge	1	1/1	0	0/1	0	0/1	1

Camilo Andres Samudio Lazaro / Ichel Alejandra Delgado Morales	Projects	4	4/4	0	0/4	0	0/4	4
Ichel Alejandra Delgado Morales	Notifications	2	2/2	0	0/2	0	0/2	2
	Pages	3	3/3	0	0/3	0	0/3	3
	Wiki	2	2/2	0	0/2	0	0/2	2

Nota. Métricas de ejecución de los casos de prueba. *Fuente.* Autoría propia

Figura 111

Métricas de ejecución



Fuente. Autoría propia

A partir del gráfico obtenido, se puede evidenciar que todos los casos de prueba propuestos para cada uno de los módulos fueron ejecutados exitosamente.

Métricas acordes a Severidad

Las métricas de severidad son aquellas en las cuales se clasifican y cuantifican los defectos o fallos encontrados en el sistema de acuerdo con su impacto en el funcionamiento y la calidad del software. Gracias a estas métricas, es posible priorizar los errores para que quienes se encargan de resolverlos puedan ocuparse primero de aquellos defectos que tienen

un impacto más crítico en el sistema o en la experiencia del usuario.

En la siguiente tabla se muestran los incidentes encontrados clasificados acorde a su severidad:

Tabla 60

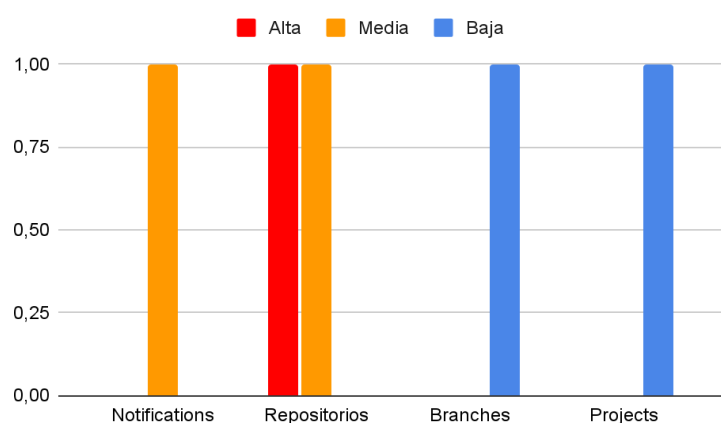
Métricas acordes a severidad

Responsable	Módulo	Alta		Media		Baja		CP Totales
		#	%	#	%	#	%	
Ichel Alejandra Delgado Morales	Notifications	0	0/2	1	1/2	0	0/2	2
Tannia Estefanny Duarte Vargas	Repositorios	1	1/3	1	1/3	0	0/3	3
Tannia Estefanny Duarte Vargas	Branches	0	0/2	0	0/2	1	1/2	2
Camilo Samudio	Projects	0	0/4	0	0/4	1	1/4	4

Nota. Métricas acordes a severidad de los casos de prueba. *Fuente.* Autoría propia

Figura 112

Métricas de severidad



Fuente. Autoría propia

A partir de este gráfico, se puede evidenciar que el módulo de repositorios fue en el que se encontraron métricas de severidad alta, por lo que deben ser abordados de manera inmediata para evitar mayores consecuencias. Por otro lado, aunque los problemas identificados en los módulos de branches y projects son menos críticos, es recomendable mantenerlos bajo observación para evitar que se acumulen y se conviertan en problemas mayores.

Métricas Acordes a Tipo de Incidencia

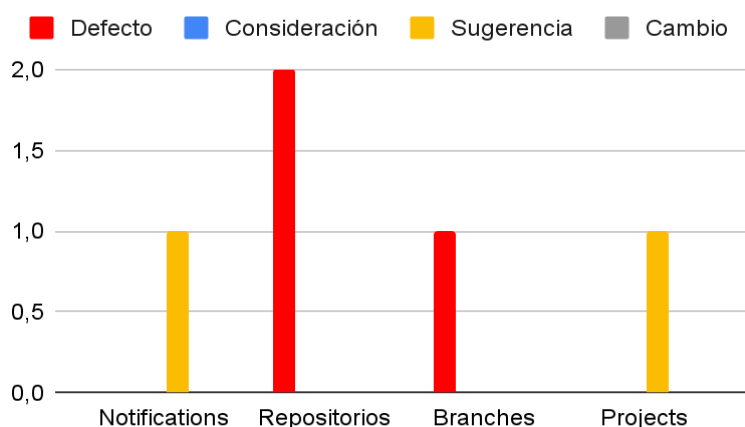
Esta clasificación permite comprender mejor la causa raíz de los errores y priorizar la resolución de problemas según su tipo. A continuación, se muestran los incidentes encontrados clasificados acorde al tipo de Incidencia:

Tabla 61

Métricas acordes al tipo de Incidencia

Responsable	Módulo	Defecto		Considerac.		Sugerencia		Cambio		CP Total
		#	%	#	#	#	%	#	%	
Ichel Alejandra Delgado Morales	Notifications	0	1/2	0	0/2	1	1/2	0	0/2	2
Tannia Estefanny Duarte Vargas	Repositorios	2	2/3	0	0/3	0	0/3	0	0/3	3
Tannia Estefanny Duarte Vargas	Branches	1	1/2	0	0/2	0	0/2	0	0/2	2
Camilo Samudio	Projects	0	0/4	0	0/4	1	1/4	0	0/4	4

Nota. Métricas acordes a tipo de incidencia de los casos de prueba. *Fuente.* Autoría propia

Figura 113*Métricas por tipo de incidencia*

Fuente. Autoría propia

A partir de estos resultados, se puede observar que el tipo de incidencia predominante fue el de defectos en el módulo de repositorios, lo cual permite evidenciar que este módulo requiere una atención inmediata y que estos defectos deben ser atendidos para así evitar otros problemas.

Métricas Acordes a su Prioridad

Estas métricas permiten determinar el orden en que deben ser abordados los problemas encontrados durante el proceso de pruebas, de modo que los defectos más críticos sean solucionados primero, optimizando el tiempo y los recursos disponibles. En la siguiente tabla se muestran los incidentes encontrados clasificados acorde a su prioridad:

Tabla 62*Métricas acordes a su prioridad*

Responsable	Módulo	Alta		Normal		Baja		CP Totales
		#	%	#	%	#	%	
Ichel Alejandra Delgado Morales	Notifications	0	0/2	1	1/2	0	0/2	2
Tannia Estefanny Duarte Vargas	Repositorios	2	2/3	0	0/3	0	0/3	3
Tannia	Branches	0	0/2	1	1/2	0	0/2	2

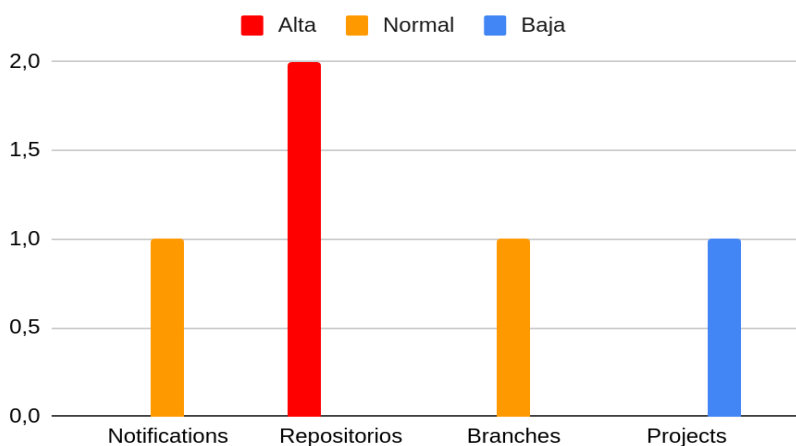
Estefanny
Duarte
Vargas
Camilo
Samudio

Projects 0 0/4 0 0/4 1 1/4 4

Nota. Métricas acordes a la prioridad de los casos de prueba. *Fuente.* Autoría propia

Figura 114

Métricas acordes a su prioridad



Fuente. Autoría propia

A partir de este gráfico, se puede evidenciar la importancia que representa el abordar los problemas encontrados en el módulo de repositorios, pues son de prioridad alta. Así mismo, se evidencia que los módulos de notifications y branches pueden ser atendidos una vez se resuelvan aquellos problemas encontrados en repositorios.

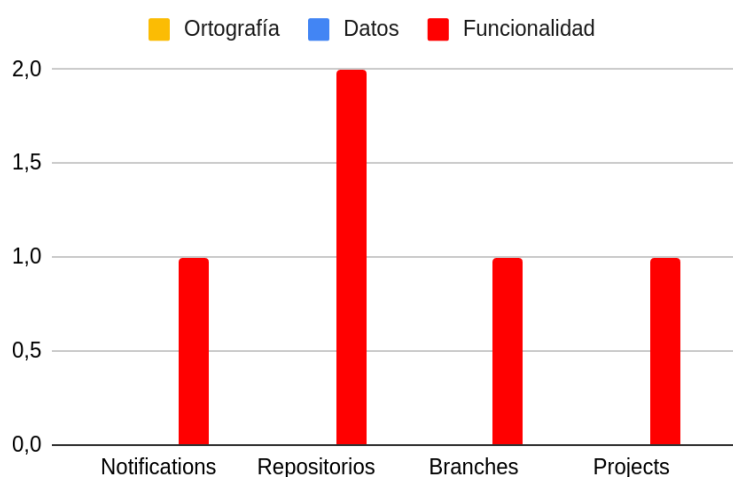
Métricas por Naturaleza

A partir de estas métricas, se pueden detectar patrones recurrentes y tomar medidas para mejorar las áreas específicas del proceso que están generando problemas, reduciendo así la cantidad y gravedad de los defectos en el futuro. En la siguiente tabla se muestran los incidentes encontrados clasificados acorde a su naturaleza, dependiendo de los diferentes tipos de naturaleza que identifiquen.

Tabla 63*Métricas acordes a su naturaleza*

Responsable	Módulo	Ortografía		Datos		FuncionalidadCP		Totales
		#	%	#	%	#	%	
Ichel Alejandra Delgado Morales	Notifications	0	0/2	0	0/2	1	1/2	2
Tannia Estefanny Duarte Vargas	Repositorios	0	0/3	0	0/3	2	2/3	3
Tannia Estefanny Duarte Vargas	Branches	0	0/2	0	0/2	1	1/2	2
Camilo Samudio	Projects	0	0/4	0	0/4	1	1/4	4

Nota. Métricas acordes a la prioridad de los casos de prueba. *Fuente.* Autoría propia

Figura 115*Métricas por naturaleza*

Fuente. Autoría propia.

A partir de los resultados obtenidos, se evidencia nuevamente que el módulo que mayor atención requiere es el de repositorios y que, además, los problemas encontrados

corresponden a funciones del sistema.

Evaluación del Producto y Proveedor

Acorde al total de defectos encontrados, se calcula el indicador de calidad del sitio y su clasificación de calidad por cada módulo así:

Tabla 64

Evaluación del producto y el proveedor

Módulo	CP ejecuta dos	#CP severidad alta	#CP severidad media	#CP severidad baja	Indicador de calidad del proveedor	Clasificación de calidad
Reposito	3	1	1	0	$1+1+0*100/$ $3 = 66.6\%$	Baja calidad
Branches	2	0	0	1	$0+0+1*100/$ $2 = 50\%$	Baja calidad

Nota. Datos obtenidos a partir de la evaluación del producto y el proveedor. *Fuente.* Autoría propia.

Informe de Pruebas

Un informe de pruebas de software es un documento esencial que resume y analiza los resultados obtenidos durante el proceso de pruebas de un sistema o aplicación. Este informe tiene como objetivo proporcionar una visión clara y objetiva sobre el estado del software, identificando las áreas de mejora, los defectos encontrados y su impacto en la calidad del producto.

Según Pressman (2020), los informes de pruebas son fundamentales para la toma de decisiones, ya que proporcionan información cuantitativa y cualitativa sobre el progreso, los riesgos y la viabilidad del software para su lanzamiento. De igual forma, Sommerville (2011) menciona que estos informes permiten comunicar de manera efectiva a los interesados los resultados de las pruebas, asegurando que se aborden las prioridades críticas antes de la entrega final del producto.

Estas bases teóricas resaltan la importancia de los informes de pruebas como herramientas clave en la gestión de calidad y en la optimización de los recursos en proyectos de desarrollo de software.

Información Básica

Fecha de inicio del proceso de pruebas: 21/AGO/2024

Fecha fin del proceso de pruebas: 26/NOV/2024

Responsables: Tannia Estefanny Duarte Vargas, Ichel Alejandra Delgado Morales,

Camilo Andres Samudio Lazaro

Alcance

Durante el proceso de pruebas, se evaluaron los siguientes módulos: Sign Up, Log In, Repositories, Issues, Pull Requests, Branches, Commits, Merge, Projects, Notifications y Pages.

Características que no fueron probadas

No se probaron las siguientes características debido a limitaciones de tiempo, alcance o recursos:

Integración con APIs externas

No se evaluó la conectividad con servicios externos como sistemas de autenticación de terceros o plataformas de despliegue continuo.

Pruebas de rendimiento y carga

No se realizaron pruebas para medir la capacidad del sistema en condiciones de alto tráfico.

Pruebas de compatibilidad en dispositivos móviles

El alcance se limitó a pruebas en navegadores web y no incluyó una evaluación exhaustiva en dispositivos móviles o tablets.

Pruebas de seguridad avanzadas

No se llevó a cabo un análisis profundo de vulnerabilidades o posibles brechas de seguridad.

Métricas de Ejecución del Proyecto

Tabla 65

Métricas de ejecución del proyecto

Módulo	Fase	Fecha inicial	Fecha Final	Tiempo estimado en horas	% avance	Fecha real inicial	Fecha real final	Tiempo en horas	% Avance real
Sign Up	Análisis	21/AGO/2024	23/AGO/2024	6	100%	21/AGO/2024	23/AGO/2024	6	100%
	Diseño	24/AGO/2024	26/AGO/2024	5	100%	24/AGO/2024	26/AGO/2024	5	100%
	Ejecución	27/AGO/2024	29/AGO/2024	7	100%	27/AGO/2024	29/AGO/2024	7	100%
Log In	Análisis	21/AGO/2024	23/AGO/2024	6	100%	21/AGO/2024	23/AGO/2024	6	100%
	Diseño	24/AGO/2024	26/AGO/2024	5	100%	24/AGO/2024	26/AGO/2024	4.5	90%
	Ejecución	27/AGO/2024	29/AGO/2024	8	100%	27/AGO/2024	29/AGO/2024	7	90%
Repositories	Análisis	30/AGO/2024	02/SEP/2024	8	100%	30/AGO/2024	02/SEP/2024	8	100%
	Diseño	03/SEP/2024	05/SEP/2024	6	100%	03/SEP/2024	05/SEP/2024	6	100%
	Ejecución	06/SEP/2024	10/SEP/2024	10	90%	06/SEP/2024	10/SEP/2024	9	90%
Issues	Análisis	11/SEP/2024	12/SEP/2024	5	100%	11/SEP/2024	12/SEP/2024	5	100%
	Diseño	13/SEP/2024	14/SEP/2024	4	100%	13/SEP/2024	14/SEP/2024	4	100%
	Ejecución	15/SEP/2024	17/SEP/2024	8	100%	15/SEP/2024	17/SEP/2024	8	100%

Pull Requests	Análisis	18/SEP/2024	20/SEP/2024	6	100%	18/SEP/2024	20/SEP/2024	6	100%
	Diseño	21/SEP/2024	23/SEP/2024	5	100%	21/SEP/2024	23/SEP/2024	5	100%
	Ejecución	24/SEP/2024	27/SEP/2024	7	100%	24/SEP/2024	27/SEP/2024	7	100%
Sign Up	Análisis	21/AGO/2024	23/AGO/2024	6	100%	21/AGO/2024	23/AGO/2024	6	100%
Branches	Análisis	28/SEP/2024	30/SEP/2024	5	100%	28/SEP/2024	30/SEP/2024	5	100%
	Diseño	01/OCT/2024	02/OCT/2024	3	100%	01/OCT/2024	02/OCT/2024	3	100%
	Ejecución	03/OCT/2024	05/OCT/2024	6	100%	03/OCT/2024	05/OCT/2024	6	100%
Commits	Análisis	06/OCT/2024	08/OCT/2024	6	100%	06/OCT/2024	08/OCT/2024	6	100%
	Diseño	09/OCT/2024	10/OCT/2024	4	100%	09/OCT/2024	10/OCT/2024	4	100%
	Ejecución	11/OCT/2024	13/OCT/2024	8	100%	11/OCT/2024	13/OCT/2024	8	100%

Nota. Datos obtenidos de las métricas de ejecución del proyecto. *Fuente.* Autoría propia

Lecciones Aprendidas

Durante el proceso de pruebas, aprendimos la importancia de una planificación clara pero flexible que nos permitió ajustar tiempos y recursos según los desafíos que surgieron. Aunque en algunos casos la ejecución no coincide completamente con lo planeado, las métricas de ejecución fueron clave para optimizar tiempos y mantenerse enfocados. El trabajo en equipo y la articulación entre los miembros fue esencial, ya que cada uno aportó habilidades diferentes que enriquecieron el resultado final.

La comunicación constante fue fundamental para resolver problemas rápidamente, tomar decisiones informadas y evitar malentendidos. También comprendimos que la retroalimentación continua y el manejo adecuado de los riesgos son claves para detectar problemas tempranamente y mejorar el proceso. En resumen, esta experiencia nos enseñó que la planificación, la comunicación efectiva y la adaptabilidad son cruciales para el éxito de cualquier proyecto.

Conclusiones

A partir de los resultados obtenidos en las métricas se puede concluir que todos los casos de prueba fueron ejecutados con éxito, lo que indica una alta efectividad en la fase de pruebas y un control adecuado de calidad en los módulos evaluados, reflejando un proceso de planificación y ejecución bien estructurado.

Los módulos de repositorios y branches presentaron defectos con severidad alta lo que subraya la necesidad de priorizar su resolución inmediata, aunque los incidentes de alta severidad fueron pocos es fundamental abordar estos problemas para evitar efectos negativos en el funcionamiento global del sistema.

Los defectos encontrados en los módulos de repositorios y projects fueron clasificados como de alta prioridad, lo que reafirma que estos módulos deben ser los primeros en recibir atención para asegurar la estabilidad y funcionalidad del software. Los defectos fueron principalmente de tipo defecto en el módulo de repositorios, lo que sugiere que este módulo requiere una revisión más exhaustiva para evitar la aparición de nuevos problemas.

La mayoría de los problemas identificados en los módulos estuvieron relacionados con la funcionalidad del sistema, lo que destaca la importancia de realizar pruebas funcionales detalladas para garantizar que todas las características operen correctamente según lo esperado.

En conclusión, las métricas reflejan un proceso de pruebas exitoso, pero también identifican áreas críticas que requieren atención es esencial, seguir realizando ajustes y mejoras para asegurar un desarrollo continuo del software con un enfoque en la calidad.

Referencias Bibliográficas

- Amórtegui, M (2022). Automatización de Pruebas con Selenium. [OVI]. Bogotá D.C., Colombia. <https://repository.unad.edu.co/handle/10596/49142>
- Blanco, F (2017). Diseño de procesos claves para el mejoramiento de la calidad en proyectos de software. La Habana, CU: D - Instituto Superior Politécnico José Antonio Echeverría. CUJAE, 2010. ProQuest ebrary. Páginas 59 - 65. <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/85824>
- Bojacá, E. (2019). Métricas de pruebas de software [OVI]. Bogotá D.C, Colombia. <https://repository.unad.edu.co/handle/10596/26559>
- Echeverri, J (2013). Reflexiones sobre Ingeniería de Requisitos y Pruebas de Software. Páginas 65 a 74. <https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/68913>
- ISTQB®, I. (2018). Programa de estudio de nivel básico ISTQB. Colombia. Páginas 89. <https://repository.unad.edu.co/handle/10596/26560>
- ISTQB®, I. (2018). Programa de estudio de nivel básico ISTQB. Colombia. Páginas 90-102. <https://repository.unad.edu.co/handle/10596/26560>
- Singh, Y. (2011). Software Testing. Cambridge University Press. Página 420-429. https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=465756&lang=es&site=eds-live&scope=site&ebv=EB&ppid=pp_420
- Sivaji, A., Razak, R. A., Mohamad, N. F., Sazali, N. (2020). Software testing automation, 7–12. <https://doi-org.bibliotecavirtual.unad.edu.co/10.1109/ICOS50156.2020.9293650>