Contexto Pruebas API-BOOKINDER

Universidad De Medellín.

Facultad de Ingeniería.

Validación y Verificación de Software.

Sara Velásquez, Diego Alejandro Durango,

Santiago Franco, Alexander Restrepo.

2024.

Control de versiones

Fecha	Actividad	Descripción	Autor	Versión
29/02/2023	Realización plan de pruebas	Plan inicial	Equipo QA Bookinder	1.0
13/03/2024	Correcciones información	Se realiza correcciones y adiciones a los planes y tipos de pruebas	Equipo QA Bookinder	2.0

Contenido

Introducción

Actualmente, existen diversas opciones de eCommerce dedicadas a la venta de libros, tanto nuevos como usados. Sin embargo, nos propusimos la creación de una plataforma que no solo integre a un extenso mercado, sino que también implemente iniciativas para simplificar las tareas tanto del vendedor al registrar libros como para establecer conexiones con los compradores a nivel nacional.

Desarrollamos un software versátil, compatible con múltiples plataformas (celular y web), que permite la integración del vendedor y el comprador de libros usados en un único espacio. Además, añadimos funcionalidades que aportan un valor agregado al producto, como la capacidad de escanear libros para registrarlos mediante sus códigos ISBN.

Este conjunto de software se divide en dos capas: el Front-end (aplicación móvil y sitio web), encargado de interactuar con el usuario y enviar las solicitudes correspondientes; y el Back-end, representado por la API BOOKINDER. Esta API, desarrollada con Spring Boot como framework y Java como lenguaje de programación, maneja solicitudes de tipo RESTful (Representational State Transfer) a través de métodos HTTP, permitiendo el envío de información en formato JSON.

En este documento, nos enfocaremos en analizar la importancia de las pruebas para la segunda capa, el Back-end, ya que constituye el corazón del conjunto de software. Evaluaremos aspectos como eficacia, eficiencia y estructuración, con el objetivo de prevenir fallos relacionados con la demanda, infracciones de seguridad e indisponibilidad en el servicio. Nuestro objetivo es garantizar el cumplimiento de los requisitos funcionales y asegurar la calidad general del sistema, basándonos en los estándares vigentes.

Glosario

- **RESTful**: El estilo de arquitectura utilizado por la API BOOKINDER, que define cómo deben ser las solicitudes y respuestas entre el cliente y el servidor.
- **Spring Boot**: El framework de Java utilizado para implementar la API BOOKINDER, que simplifica el desarrollo de aplicaciones Java.
- **CRUD:** El acrónimo que representa las operaciones básicas de la API BOOKINDER: Crear, Leer, Actualizar y Borrar datos.
- **TOKEN**: La cadena de caracteres utilizada para autenticar y autorizar las transacciones realizadas a través de la API BOOKINDER
- **MVC**: La arquitectura de software utilizada por la API BOOKINDER para separar la lógica de la aplicación en modelos, vistas y controladores.
- **Spring Security:** El módulo de Spring utilizado para proporcionar seguridad y protección a la API BOOKINDER, evitando accesos no autorizados.
- **Base de Datos:** El sistema utilizado para almacenar los datos de usuarios, libros, categorías, etc., que son gestionados por la API BOOKINDER.
- **JSON:** El formato utilizado para enviar y recibir datos entre el Front-end y el Back-end de la aplicación a través de la API BOOKINDER.
- **Login**: Acceso de usuarios a la plataforma de la API BOOKINDER mediante nombre de usuario y contraseña.
- **Framework**: es una estructura conceptual y tecnológica que proporciona un conjunto de herramientas, bibliotecas y reglas que facilitan el desarrollo de software.
- **Front-End**: El término "Front-end" se refiere a la parte de un sistema o aplicación informática que interactúa directamente con los usuarios finales. En el desarrollo de software, especialmente en el desarrollo web, se distingue entre el Front-end y el back-end.
- **Back-End:** El "back-end" se refiere a la parte de un sistema informático o aplicación que no es visible para los usuarios finales, pero que realiza funciones esenciales para el correcto funcionamiento del software.
- **Endpoints:** En el contexto de servicios web, un "endpoint" se refiere a un punto final de una comunicación, ya sea para obtener o enviar información. En términos más simples, un endpoint es una URL específica a la que se puede enviar una solicitud, y generalmente se asocia con un recurso o una operación en una API (Interfaz de Programación de Aplicaciones).
- **E-eCommerce:** El término "eCommerce" se refiere a la compra y venta de bienes y servicios a través de internet. Es la abreviatura de "comercio electrónico". El eCommerce implica realizar transacciones comerciales electrónicas, donde los compradores y vendedores participan en intercambios comerciales a través de plataformas en línea.
- **Cross-site scripting (XSS):** Este tipo de XSS ocurre cuando la manipulación de documentos en el lado del cliente (DOM) es controlada por el script malicioso, y no por el servidor. El ataque ocurre en el navegador del usuario después de que la página ha sido entregada.

Descripción & Contexto del software

La API BOOKINDER sigue la arquitectura MVC (Modelo Vista Controlador), y está implementada en Java con el framework Spring Boot. Esta API se consume a través de endpoints que utilizan los métodos HTTP, permitiendo el envío de información en formato JSON. Facilita operaciones de Agregar, Eliminar, Actualizar y Consultar (CRUD) datos relacionados con los usuarios de la aplicación (vendedores, compradores o administradores), así como con los libros almacenados en la base de datos, los cuales se comercializan en la plataforma junto con sus diversas categorías y características.

La API cuenta con un mecanismo de seguridad mediante tokens para las transacciones. Estos tokens deben ser generados y utilizados por usuarios con credenciales válidas registradas en la base de datos, garantizando así la seguridad durante las transacciones. Además, se ha implementado una configuración de Spring Security para evitar el acceso no autorizado a los endpoints. También se incluye una página de bienvenida que valida la disponibilidad de la API en el servidor. Esta combinación de medidas asegura un entorno seguro y protegido para las operaciones realizadas a través de la API BOOKINDER.

Módulos de Software

Requisitos Funcionales

1. Conjunto Software

Los siguientes son los requisitos funcionales del conjunto software:

- El sistema ofrecerá la posibilidad de registrar un libro de manera eficiente, ya sea a través de su ISBN con integración automática de datos o mediante la opción de ingreso manual.
- Los usuarios disfrutarán de la facilidad de visualizar los libros registrados y disponibles para la compra, ya sea porque ellos mismos los registraron o porque están disponibles en el sistema.
- Tanto compradores como vendedores podrán hacer un seguimiento detallado de los libros, tanto los que han vendido como los disponibles para la venta.
- El usuario tendrá a su disposición herramientas avanzadas para buscar y filtrar libros según diversas categorías, como nombre, calificación, estado, precio, autor, entre otros.
- Además, se implementará un sistema de logs y seguimientos que permitirá realizar estadísticas, identificar errores y realizar copias de seguridad seguros, asegurando una trazabilidad efectiva del uso de las aplicaciones.
- Con un enfoque en la ética y la seguridad, el sistema incorporará un filtro eficaz para detectar y prevenir contenido inapropiado o ilegal.
- La estructura del sistema se dividirá en dos fases específicas para compradores y vendedores, y una tercera para el Administrador de plataforma, asegurando una experiencia personalizada y eficiente para cada tipo de usuario.
- En términos de gestión de usuarios, el sistema facilitará el registro y login de usuarios (compradores y vendedores), además de brindar opciones para el cierre de sesión y la recuperación de cuentas y contraseñas.

- Para resguardar la seguridad y la privacidad de la información, el sistema incorporará medidas de seguridad robustas en almacenamiento de datos e información de uso, adaptándose plenamente a las normativas vigentes.

2. Back-end API BOOKINDER

Basados en estos requisitos se procede a generar los requisitos del API BOOKINDER respectivamente:

- En el proceso de registro de libros, el API requerirá de manera obligatoria un ISBN válido dentro del campo JSON. Si no es coherente con el nombre, el sistema podría evitar el registro por contenido inapropiado o ilegal.
- La API proporcionará todos los métodos CRUD, tanto para libros como para usuarios, garantizando una gestión completa de la información.
- Deberá ofrecer un endpoint específico para la búsqueda de ISBN, integrado con otras plataformas como Google Books, mejorando la capacidad de obtener información precisa y detallada.
- La API incluirá una página que validará su disponibilidad en el servidor, asegurando un acceso fácil y rápido para los usuarios.
- Se implementará un método que permitirá consultar los libros por comprador y su estado de venta (vendido o disponible), brindando una visión clara del historial de transacciones.
- La búsqueda de libros será altamente personalizable, permitiendo filtros según diversas categorías como nombre, calificación, estado, precio, autor, entre otros.
- La API garantizará un proceso de login seguro, generando un token de transacción válido por un tiempo establecido, contribuyendo a la seguridad del sistema.
- Solo se permitirá el consumo de la API desde direcciones IP previamente establecidas, reforzando la seguridad y limitando el acceso a fuentes autorizadas.
- Se implementará un sistema de logs detallado que registre transacciones y errores, asegurando la trazabilidad de las operaciones sin comprometer datos importantes en este rastro.

Plan de pruebas

1. Descripción

En este plan de pruebas, se abordarán las distintas áreas del API, clasificadas de la siguiente manera:

- **Pruebas Unitarias:** Se llevarán a cabo pruebas exhaustivas a nivel de unidades individuales de código para asegurar que cada componente funcione correctamente de manera aislada. Esto incluirá la validación de funciones, métodos y clases a nivel de código.
- **Pruebas de Funcionalidad:** En esta categoría, se evaluará el comportamiento funcional del API en su conjunto. Se verificará la correcta ejecución de las funciones específicas requeridas por la lógica de negocio, asegurando que el API cumpla con los requisitos funcionales establecidos.
- **Pruebas de Integración:** Se realizarán pruebas para evaluar la interacción y cooperación entre los distintos módulos del API. Esto incluirá la validación de la correcta comunicación entre componentes y la integridad del flujo de datos a través de las interfaces.
- **Pruebas de Seguridad:** Esta área se centrará en evaluar la robustez del API frente a posibles vulnerabilidades y amenazas de seguridad. Se llevarán a cabo pruebas de penetración para identificar posibles brechas de seguridad y se verificará el cumplimiento de las medidas de protección implementadas.
- **Pruebas de Rendimiento:** Estas pruebas se centran en evaluar cómo responde la API bajo diferentes niveles de carga. El objetivo es medir el rendimiento y la capacidad de respuesta del sistema, especialmente en condiciones de alta demanda. Se realizan pruebas de carga para simular un gran volumen de transacciones y así determinar la escalabilidad del sistema.
- **Pruebas de Estrés:** Estas pruebas evalúan la resistencia de la API bajo condiciones extremas de carga o situaciones adversas. El objetivo es identificar posibles puntos de fallo o degradación del rendimiento cuando la API se enfrenta a cargas inesperadas o sobrecargas.
- **Pruebas de Usabilidad:** Estas pruebas se centran en evaluar la facilidad de uso y la experiencia del usuario al interactuar con la API. El objetivo es validar aspectos como el registro de usuarios, el proceso de inicio de sesión, así como la eficacia de las herramientas de búsqueda y filtrado.
- **Pruebas de Regresión:** Estas pruebas tienen como objetivo garantizar que las nuevas actualizaciones o cambios en la API no afecten negativamente a las funcionalidades existentes. Se vuelven a ejecutar pruebas unitarias y de funcionalidad después de cada modificación en el código o actualización de la API para asegurar la integridad del sistema.

Cada área de pruebas se diseñará de manera específica para abordar los aspectos clave relacionados con la calidad y el rendimiento del API. La combinación de estas pruebas proporcionará una cobertura integral, garantizando la fiabilidad y la seguridad del sistema en su conjunto.

2. ¿Qué?, ¿Por qué? y ¿Para qué?

En este plan de pruebas, se abordarán las distintas áreas del API, clasificadas de la siguiente manera:

- **Pruebas Unitarias:**
  Porque hay que validar cada funcionalidad de código independiente del flujo completo de la lógica de negocio, necesitamos aislar de la manera más eficiente y práctica posible las funcionalidades requeridas por el cliente dentro de nuestras pruebas.
  Para poder probar de manera independiente cómo funcionan en nuestra función de código cada escenario de prueba, ya sea el correcto funcionamiento o errores esperados y si no se realizan las correcciones, se hace con ánimo de que la funcionalidad mantenga su correcto funcionamiento a largo plazo y no sufra alteraciones de otros desarrollos.

- **Pruebas de Funcionalidad:**
  Porque es necesario realizar validaciones pertinentes sin tener conocimiento interno del funcionamiento del software como están funcionando los pequeños componentes que conforman un caso de uso.
  Para asegurar la calidad de un desarrollo externo, generando escenarios de prueba basados en los casos de uso presentados por el cliente, para evitar que posibles bugs lleguen a los ambientes productivos, normalmente se capturan en ambientes destinados a realizar pruebas, aunque se encuentren en etapa de desarrollo para evitar costos.

- **Pruebas de Integración:**
  Porque así mismo como se revisan las funcionalidades a bajo nivel como desarrollador y posteriormente a más alto nivel ya sea como QA o como Dev estas pruebas se hacen a mucho más alto nivel, probando como se está comportando la interacción entre los diferentes componentes que pueden estar presentes en nuestros aplicativos, ya sean internos o externos a nuestra compañía.
  Para de la siguiente manera poder asegurarnos de que tanto la información que se está enviando desde nuestro aplicativo hacia otros está saliendo de una manera correcta, se está generando la respuesta adecuada y/o esperada, así mismo también podremos validar como se está recibiendo y procesando la información que recibimos de servicios externos.

- **Pruebas de Seguridad:**
  Las pruebas de seguridad en aplicativos webs se realizan para identificar y mitigar vulnerabilidades que podrían ser explotadas por atacantes, garantizando así la integridad, confidencialidad y disponibilidad de la información.
  Estas pruebas buscan proteger contra ataques como inyecciones de código, Cross-site scripting (XSS) y otros riesgos de seguridad, fortaleciendo la resistencia del aplicativo a posibles amenazas, previniendo fugas de datos y asegurando la confianza de los usuarios. La seguridad proactiva a través de pruebas contribuye a un entorno digital más confiable y protegido contra posibles ciberataques.

- **Pruebas de Rendimiento:**
  Qué: Evaluar el rendimiento y la capacidad de respuesta de la API bajo diferentes condiciones de carga.
  Por qué: Para asegurarse de que la API pueda manejar eficientemente un alto volumen de transacciones y determinar su escalabilidad.
  Para qué: Garantizar una experiencia de usuario fluida y evitar problemas de rendimiento o caídas del sistema cuando se enfrenta a cargas pesadas.

- **Pruebas de Estrés:**
  Qué: Evaluar cómo la API responde bajo condiciones extremas de carga o situaciones adversas.
  Por qué: Identificar posibles puntos de fallo o degradación del rendimiento cuando la API está sometida a cargas inesperadas o sobrecargas.
  Para qué: Mejorar la fiabilidad y la resistencia de la API, asegurando que pueda mantenerse estable incluso en situaciones críticas.

- **Pruebas de Usabilidad:**
  Qué: Evaluar la facilidad de uso y la experiencia del usuario al interactuar con la API.
  Por qué: Validar aspectos como el registro de usuarios, el inicio de sesión y la eficacia de las herramientas de búsqueda y filtrado para mejorar la experiencia del usuario.
  Para qué: Asegurar que la API sea intuitiva y fácil de usar, lo que puede aumentar la satisfacción del usuario y la adopción del producto.

- **Pruebas de Regresión:**
  Qué: Garantizar que las nuevas actualizaciones o cambios no afecten negativamente a las funcionalidades existentes.
  Por qué: Para asegurar la integridad del sistema y evitar la introducción de errores o problemas durante el desarrollo de la API.
  Para qué: Mantener la estabilidad y la consistencia de la API a lo largo del tiempo, asegurando que las nuevas versiones no rompan funcionalidades previamente implementadas.

3. Objetivos

Pruebas unitarias

- Verificar que cada función de registro de libros a través del ISBN funcione correctamente
- Validar que los libros se registren correctamente en la base de datos.
- Validar que se generen los registros adecuados en el log de transacciones.

Pruebas de funcionalidad:

- Hay que confirmar que la funcionalidad de búsqueda de libros de resultados precisos
- Validar que los resultados de búsqueda sean coherentes con los criterios ingresados

Pruebas de integración:

- Garantizar la correcta integración entre el componente de registro de libros y la base de datos
- Comprobar la coherencia entre la información registrada y los datos proporcionados
- Probar la resistencia frente a intentos de acceso de autorizados
- Identificar las vulnerabilidades del API.

Pruebas de Rendimiento:

- Evaluar el rendimiento y la capacidad de respuesta de la API bajo diferentes condiciones de carga para garantizar que pueda manejar eficientemente un alto volumen de transacciones.
- Identificar cuellos de botella y puntos de congestión en el sistema para mejorar la escalabilidad y la eficiencia de la API.

Pruebas de Estrés:

- Evaluar la resistencia de la API bajo condiciones extremas de carga o situaciones adversas para asegurar su estabilidad y fiabilidad en circunstancias críticas.
- Identificar posibles puntos de fallo o degradación del rendimiento cuando la API es sometida a cargas inesperadas o sobrecargas.

Pruebas de Usabilidad:

- Evaluar la facilidad de uso y la experiencia del usuario al interactuar con la API para mejorar la satisfacción del usuario y la adopción del producto.
- Validar aspectos como el registro de usuarios, el inicio de sesión y la eficacia de las herramientas de búsqueda y filtrado para garantizar una experiencia de usuario intuitiva.

Pruebas de Regresión:

- Garantizar que las nuevas actualizaciones o cambios no afecten negativamente a las funcionalidades existentes para mantener la estabilidad y la consistencia de la API a lo largo del tiempo.
- Detectar y corregir posibles regresiones o problemas introducidos durante el desarrollo de la API para asegurar su integridad y funcionalidad continua.

4. Alcance

Operaciones CRUD
Evaluación de la API BOOKINDER en términos de eficacia y consistencia en la ejecución de operaciones de Agregar, Eliminar, Actualizar y Consultar datos de usuarios y libros en la base de datos.
Ejecución exitosa de operaciones CRUD sin pérdida de datos ni inconsistencias.

Rendimiento
Pruebas de carga y estrés para garantizar que la API pueda manejar la carga esperada sin degradación significativa del rendimiento.
Seguridad
Análisis exhaustivo de la implementación de seguridad, incluyendo la validación de tokens, restricciones de acceso y protección contra posibles

5. Enfoque

Pruebas unitarias

- Se desarrollan casos de prueba para cada funcionalidad de código independiente.
- Se realizan pruebas de regresión para garantizar el correcto funcionamiento a largo plazo.

Pruebas funcionales

- Se generan escenarios de prueba basados en casos de uso proporcionados por el cliente.
- Se capturan y corrigen bugs antes de que lleguen a los ambientes productivos

Pruebas de integración

- Se valida el envío y recepción de información hacia y desde otros servicios
- Se verifica la generación y procesamiento adecuado de respuestas entre los componentes

Pruebas de seguridad

- Se realizan pruebas de penetración para identificar posibles vulnerabilidades
- Se evalúa la configuración del servidor y del aplicativo en busca de posibles puntos débiles
- Ausencia de vulnerabilidades identificadas durante pruebas de seguridad y cumplimiento de prácticas de seguridad recomendadas.

Pruebas de Rendimiento:

- Realizar pruebas de carga para simular un alto volumen de transacciones y evaluar la escalabilidad del sistema.
- Medir y analizar los tiempos de respuesta de la API bajo diferentes niveles de carga para identificar posibles cuellos de botella.
- Utilizar herramientas de prueba de rendimiento para generar carga y monitorear el comportamiento del sistema en tiempo real.

Pruebas de Estrés:

- Someter la API a condiciones de sobrecarga para identificar posibles puntos de fallo o degradación del rendimiento.
- Incrementar gradualmente la carga sobre la API hasta alcanzar su límite de capacidad para determinar su resistencia y estabilidad.
- Observar cómo responde la API ante situaciones extremas y cómo se recupera después de períodos de alta carga.

Pruebas de Usabilidad:

- Validar la facilidad de registro y login de usuarios, así como la eficacia de las herramientas de búsqueda y filtrado.
- Realizar pruebas de usabilidad con usuarios reales o grupos de prueba para obtener retroalimentación sobre la experiencia de usuario.
- Identificar áreas de la interfaz de usuario que pueden ser confusas o difíciles de usar y proponer mejoras para optimizar la experiencia del usuario.

Pruebas de Regresión:

- Re-ejecutar pruebas unitarias y de funcionalidad después de cada cambio en el código o actualización de la API para detectar posibles regresiones.
- Utilizar técnicas de automatización de pruebas para garantizar una cobertura exhaustiva de todas las funcionalidades afectadas por el cambio.
- Comparar los resultados de las pruebas con versiones anteriores de la API para asegurar que no se hayan introducido errores o cambios no deseados.

6. Estrategias de Validación & Verificación

Validación

- **Revisión por pares**: Se involucra a otros desarrolladores para revisar el código y las funcionalidades.
- **Revisión por cliente:** los clientes prueban las funcionalidades para validar que cumplan con los requerimientos.

Verificación

Pruebas unitarias:

- Desarrollar casos de prueba para cada funcionalidad de código independiente.
- Ejecutar pruebas para verificar el correcto funcionamiento y manejo de errores.
- Realizar pruebas de regresión para garantizar el correcto funcionamiento a largo plazo.

Pruebas funcionales:

- Generar escenarios de prueba basados en casos de uso presentados por el cliente.
- Ejecutar pruebas para validar el comportamiento del software desde una perspectiva externa.
- Capturar y corregir bugs antes de que lleguen a los ambientes productivos.

Pruebas de integración:

- Probar la interacción entre diferentes componentes internos y externos del aplicativo.
- Validar el envío y recepción de información hacia y desde otros servicios.
- Verificar la generación y procesamiento adecuado de respuestas entre los componentes.

Pruebas de seguridad:

- Identificar y mitigar vulnerabilidades como inyecciones de código y Cross-site scripting.
- Garantizar la integridad, confidencialidad y disponibilidad de la información.
- Prevenir fugas de datos y proteger contra posibles ciberataques.

Pruebas de regresión:

- Se ejecutan para asegurar que las nuevas actualizaciones no hayan afectado funciones existentes.

7. Responsables de Pruebas y Descripción de Usuarios:

Responsables

Líder de pruebas (Test Lead):

- Coordinará y supervisará todas las actividades de pruebas para API BOOKINDER.
- Sera el responsable de la planificación, diseño y ejecución de pruebas.
- Rexportará y gestionará los problemas identificados durante las pruebas.

Ingeniero de pruebas (Test Engineer):

- Creará casos de prueba detallados para cubrir todos los aspectos de la API.
- Ejecutará pruebas unitarias, de funcionalidad, integración y seguridad.
- Automatizara pruebas cuando sea posible para mejorar la eficacia del proceso.

Analista de pruebas (Test Analyst):

- Analizara los requisitos y documentara escenarios de prueba.
- Colaborará con los desarrolladores y diseñadores para comprender el comportamiento esperado de la API.
- Participara en la identificación y priorización de defectos.

Especialista de seguridad de TI (IT Security Specialist):

- Se centrará en las pruebas de seguridad de la API BOOKINDER.
- Realizara pruebas de penetración para identificar posibles vulnerabilidades.
- Asegurara que se cumplan los estándares de seguridad establecidos.

Usuarios:

Administrador de plataforma:

- Gestiona la configuración general de la API BOOKINDER.
- Tiene acceso a las herramientas de administración y estadísticas.
- Puede realizar operaciones CRUD en usuarios y libros.

Vendedor:

- Registra libros para la venta a través de la API.
- Realiza un seguimiento de los libros disponibles y vendidos.
- Utiliza herramientas de búsqueda y filtros para gestionar su inventario.

8. Tipos

Pruebas Unitarias:

Objetivo: Verificar que cada unidad individual de código funcione como se espera.

Enfoque para API BOOKINDER: Evaluar funciones y métodos específicos de la API a nivel de código.

Pruebas de Funcionalidad:

Objetivo: Evaluar el comportamiento funcional de la API según los requisitos.

Enfoque para API BOOKINDER: Verificar la correcta ejecución de funciones específicas como el registro de libros, la gestión de usuarios, la búsqueda de libros, etc.

Pruebas de Integración:

Objetivo: Evaluar la interacción y cooperación entre los distintos módulos de la API.

Enfoque para API BOOKINDER: Validar la comunicación entre la gestión de usuarios y la gestión de libros, asegurando la integridad del flujo de datos.

Pruebas de Seguridad:

Objetivo: Evaluar la robustez del API frente a posibles vulnerabilidades y amenazas de seguridad.

Enfoque para API BOOKINDER: Probar la generación y validación segura de tokens, acceso desde direcciones IP autorizadas, y realizar pruebas de penetración para identificar posibles brechas de seguridad.

Pruebas de Rendimiento:

Objetivo: Evaluar el rendimiento y la capacidad de respuesta de la API bajo diferentes condiciones de carga.

Enfoque para API BOOKINDER: Realizar pruebas de carga para simular un alto volumen de transacciones y evaluar la escalabilidad del sistema.

Pruebas de Estrés:

Objetivo: Evaluar cómo la API responde bajo condiciones extremas de carga o situaciones adversas.

Enfoque para API BOOKINDER: Someter la API a condiciones de sobrecarga para identificar posibles puntos de fallo o degradación del rendimiento.

Pruebas de Usabilidad:

Objetivo: Evaluar la facilidad de uso y la experiencia del usuario al interactuar con la API.

Enfoque para API BOOKINDER: Validar la facilidad de registro y login de usuarios, así como la eficacia de las herramientas de búsqueda y filtrado.

Pruebas de Regresión:

Objetivo: Garantizar que las nuevas actualizaciones o cambios no afecten negativamente a las funcionalidades existentes.

Enfoque para API BOOKINDER: Re ejecutar pruebas unitarias y de funcionalidad después de cada cambio en el código o actualización de la API.

Pruebas de Trazabilidad:

Objetivo: Asegurar que cada requisito funcional esté cubierto por las pruebas correspondientes.

Enfoque para API BOOKINDER: Relacionar cada caso de prueba con los requisitos funcionales establecidos para la API.

9. Técnicas

Pruebas de Caja Negra:

Descripción: Se prueba la API sin conocer la estructura interna del código. Los casos de prueba se diseñan en función de las especificaciones y requisitos.

Aplicación en API BOOKINDER: Crear casos de prueba basados en los requisitos funcionales para probar la entrada y salida esperada de la API.

Pruebas de Caja Blanca:

Descripción: Se examina la estructura interna del código de la API. Los casos de prueba se diseñan en función de la lógica interna de la API.

Aplicación en API BOOKINDER: Evaluar el código de la API BOOKINDER para asegurar una cobertura completa y detectar posibles caminos de ejecución.

Pruebas de Regresión:

Descripción: Se realizan pruebas para asegurar que las nuevas actualizaciones o cambios no afecten negativamente a las funcionalidades existentes.

Aplicación en API BOOKINDER: Re ejecutar casos de prueba específicos después de cada modificación para verificar que no haya regresiones en la funcionalidad.

Pruebas de Mutación:

Descripción: Introducir cambios deliberados en el código fuente (mutaciones) y ejecutar casos de prueba para asegurar que los cambios sean detectados.

Aplicación en API BOOKINDER: Introducir mutaciones en el código de la API y ejecutar pruebas para confirmar que los casos de prueba identifiquen las mutaciones.

Pruebas de Estrés:

Descripción: Evaluar cómo la API responde bajo condiciones extremas de carga o situaciones adversas.

Aplicación en API BOOKINDER: Someter la API a cargas intensas para evaluar su rendimiento y capacidad de manejar picos de tráfico.

Pruebas de Seguridad:

Descripción: Evaluar la robustez del API frente a posibles vulnerabilidades y amenazas de seguridad.

Aplicación en API BOOKINDER: Realizar pruebas de penetración, verificando la seguridad de la generación y validación de tokens, y garantizando el acceso seguro desde IP autorizadas.

Pruebas de Interfaz:

Descripción: Verificar la correcta interacción entre la API y otras aplicaciones o servicios con los que se integra.

Aplicación en API BOOKINDER: Evaluar la comunicación entre la API y otras plataformas, como Google Books, especialmente en el endpoint de búsqueda de ISBN.

Pruebas de Usabilidad:

Descripción: Evaluar la facilidad de uso y la experiencia del usuario al interactuar con la API.

Aplicación en API BOOKINDER: Validar la facilidad de registro y login de usuarios, así como la eficacia de las herramientas de búsqueda y filtrado.

Pruebas de Trazabilidad:

Descripción: Asegurar que cada requisito funcional esté cubierto por las pruebas correspondientes.

Aplicación en API BOOKINDER: Relacionar cada caso de prueba con los requisitos funcionales establecidos para la API.

10. Datos

Pruebas Unitarias:

- **Registro de Libros:**
  - Caso de prueba 1: Registrar un libro con un ISBN válido.
  - Caso de prueba 2: Intentar registrar un libro sin un ISBN válido.
  - Gestión de Usuarios:
  - Caso de prueba 3: Crear un nuevo usuario vendedor.
  - Caso de prueba 4: Actualizar la información de un usuario comprador.
  - Búsqueda de ISBN:
  - Caso de prueba 5: Realizar una búsqueda de ISBN exitosa.
  - Caso de prueba 6: Probar la búsqueda de ISBN con un ISBN no existente.

Pruebas de Funcionalidad:

- **Seguimiento de Libros:**
  - Caso de prueba 7: Comprobar la capacidad de los compradores para realizar un seguimiento detallado de los libros vendidos y disponibles.
  - Caso de prueba 8: Verificar la capacidad de los vendedores para hacer un seguimiento detallado de los libros disponibles para la venta.
  - Filtros de Búsqueda:
  - Caso de prueba 9: Utilizar filtros para buscar libros por autor.
  - Caso de prueba 10: Filtrar libros por precio y verificar la precisión de los resultados.
  - Login Seguro:
  - Caso de prueba 11: Iniciar sesión con credenciales válidas y verificar la generación de un token de transacción.

Pruebas de Integración:

- **Interacción entre Módulos:**
  - Caso de prueba 12: Verificar la correcta comunicación entre la gestión de usuarios y la gestión de libros.
  - Caso de prueba 13: Validar la integridad del flujo de datos entre la API y la base de datos.

Pruebas de Seguridad:

- **Generación de Tokens:**
  - Caso de prueba 14: Probar la generación de tokens con credenciales inválidas.
  - Caso de prueba 15: Verificar la validez y el tiempo de expiración de un token generado.
- **Acceso desde IP Autorizada:**
  - Caso de prueba 16: Intentar consumir la API desde una dirección IP no autorizada.
  - Caso de prueba 17: Verificar que la API solo sea accesible desde las direcciones IP establecidas.

11. Criterios de aceptación, criterios de entrada y salida

Código: B-001.

Título: Registro de libros.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.

Criterios de aceptación:

- **YO COMO:** Vendedor de libros.
- **QUIERO**: Que la plataforma de Bookinder se encuentre habilitada para ingresar el libro que yo quiera.
- **CUANDO:** Requiera ingresar el libro que yo requiera vender.
- **ENTONCES:** Visualizar correctamente el proceso de guardado.

Pruebas:

- **Escenario 01: Registro de Libros Con token Vencido**
- **Descripción:** Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de manera errónea los datos del Token y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con token erróneo.

Datos importantes para la prueba: Token invalidó.

- **Escenario 02: Registro de Libros con datos de un libro existente**
- **Descripción:** Se debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos de un libro ya existe y visualizando las correspondientes validaciones de la plataforma que NO PERMITA registrar el libro con datos de un libro existente.

Datos importantes para la prueba: Datos de un libro existente (sellerUser, quality, price, quantity, observations, onSale, isbn, name, author, category, description, urlImage.), token valido.

- **Escenario 03: Registro de Libros con datos de un libro Nuevo**
- **Descripción** Ve debe de realizar el proceso manual de ingresar un libro a la plataforma de Bookinder, ingresando los datos correspondientes que solicita el servicio bajo sus correspondientes parámetros y validaciones de manera correcta y como resultado visualizar el libro correctamente registrado y con los datos correctos.

Datos importantes para la prueba: Datos de un libro Nuevo (sellerUser, quality, price, quantity, observations, onSale, isbn, name, author, category, description, urlImage.), token valido.

Código: B-002.

Título: Actualizar Libro.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual actualizar un libro

Criterios de aceptación:

- **YO COMO:** Vendedor de libros.
- **QUIERO**: Que la plataforma de Bookinder se encuentre habilitada para actualizar libros.
- **CUANDO:** Requiera actualizar el libro que deseo vender
- **ENTONCES:** Visualizar correctamente el proceso de actualización.

Pruebas:

- **Escenario 01: Actualizar Libro con datos de un libro inexistente**
- **Descripción:** Se debe de realizar el proceso manual de actualizar un libro a la plataforma de Bookinder, ingresando los datos de un libro inexiste y visualizando las correspondientes validaciones de la plataforma que NO PERMITA actualizar el libro ya que este no existe, dando su respectivo dato de error

Datos importantes para la prueba: Datos de un libro inexistente (sellerUser, quality, price, quantity, observations, onSale, isbn, name, author, category, description, urlImage.), token valido.

- **Escenario 02: Actualizar Libro con datos de un libro existente**
- **Descripción:** Se debe de realizar el proceso manual de actualizar un libro a la plataforma de Bookinder, ingresando los datos de un libro que existe y visualizando las correspondientes validaciones de la plataforma que PERMITA actualizar el libro ya que este existe, dando su respectivo dato de confirmación.

Datos importantes para la prueba: Datos de un libro existente (sellerUser, quality, price, quantity, observations, onSale, isbn, name, author, category, description, urlImage.), token valido.

Código: B-003.

Título: Buscar todos los Libros.

Usuarios implicados: Vendedor-Usuario.

Yo como vendedor y/o usuario quiero ingresar a la plataforma de Bookinder y de forma manual y buscar todos los libros disponibles.

Criterios de aceptación:

- **YO COMO:** Vendedor /Usuario de Bookinder.
- **QUIERO**: Que la plataforma de Bookinder se encuentre habilitada para buscar todos los libros.
- **CUANDO:** Requiera buscar los Libros que se encuentren disponibles
- **ENTONCES:** Visualizare correctamente el proceso de Búsqueda de todos los libros.

Pruebas:

- **Escenario 01: Buscar todos los libros existentes.**
- **Descripción:** Se debe de realizar el proceso manual de buscar todos los libros de la plataforma de ApiBookKinder, visualizando las correspondientes validaciones de la plataforma que PERMITA Buscar los libros,dando su respectivo dato de confirmación.

Datos importantes para la prueba: token valido, Endpoint Disponible

Código: B-004.

Título: Eliminar Libro.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.

Criterios de aceptación:

- **YO COMO:**
- **QUIERO**:
- **CUANDO:**
- **ENTONCES:**

Pruebas:

- **Escenario 01:**

Datos importantes para la prueba:

Código: B-005.

Título: Buscar Información libro por ISBN.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y de forma manual ingresar un libro para la venta en la plataforma.

Criterios de aceptación:

- **YO COMO:**
- **QUIERO**:
- **CUANDO:**
- **ENTONCES:**

Pruebas:

- **Escenario 01:**

Datos importantes para la prueba:

Código: P-002.

Título: Registro de libros con ISBN.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y utilizando el código ISBN ingresar un libro para la venta en la plataforma.

Criterios de aceptación:

- **YO COMO:** vendedor de libros.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para registrar el libro que yo quiera.
- **CUANDO:** Requiera registrar el libro utilizando el código ISBN para recuperar los datos del libro.
- **ENTONCES:** Visualizar correctamente el proceso de guardado.

Pruebas:

- Se debe de realizar el proceso de ingresar un libro a la plataforma de Bookinder utilizando el código ISBN correcto para recuperar sus datos y como resultado visualizar el libro correctamente registrado y con los datos correctos.

Datos importantes para la prueba: ISBN del libro, precio, cantidad y estado del libro.

Código: P-003.

Título: Registro de libros con ISBN no encontrado.

Usuarios implicados: Vendedor.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y utilizando un código ISBN incorrecto.

Criterios de aceptación:

- **YO COMO:** vendedor de libros.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para registrar el libro que yo quiera.
- **CUANDO:** Requiera registrar el libro utilizando un código ISBN incorrecto para recuperar los datos del libro.
- **ENTONCES:** Visualizar las advertencias y/o errores del proceso de guardado.

Pruebas:

- Se debe de realizar el proceso de ingresar un libro a la plataforma de Bookinder utilizando un código ISBN incorrecto para recuperar sus datos y como resultado visualizar las advertencias y/o errores correspondientes al fallo de la recuperación de la información por medio del ISBN.

Datos importantes para la prueba: ISBN del libro, precio, cantidad y estado del libro.

Código: P-004.

Título: Visualizar libros a la venta.

Usuarios implicados: Vendedores y Compradores.

Yo como vendedor y como comprador quiero ingresar a la plataforma de Bookinder y visualizar mis libros en venta y los de otros vendedores.

Criterios de aceptación Vendedor:

- **YO COMO:** Vendedor de libros.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para mi ingreso como vendedor.
- **CUANDO:** Requiera visualizar mis libros en venta y los de otros vendedores.
- **ENTONCES:** Visualizar los libros que tenga en venta y los de otros vendedores.

Pruebas:

- Se debe de ingresar a la plataforma Bookinder para visualizar los libros en venta por parte del vendedor y de los otros vendedores, los libros deben de visualizarse de manera correcta en formato de lista.

Datos importantes para la prueba: Lista de libros, credenciales de usuario tipo vendedor.

Criterios de aceptación comprador:

- **YO COMO:** Comprador.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para mi ingreso como comprador.
- **CUANDO:** Requiera visualizar libros en venta publicados por otros vendedores.
- **ENTONCES:** Visualizar la lista de libros que se encuentran actualmente publicados.

Pruebas:

- Se debe de ingresar a la plataforma Bookinder como COMPRADOR para visualizar los libros en venta de los vendedores, los libros deben de visualizarse de manera correcta en formato de lista.

Datos importantes para la prueba: Lista de libros, credenciales de usuario tipo comprador.

Código: P-011.

Título: Escanear ISBN del libro para registrarlo.

Usuarios implicados: Vendedores.

Yo como vendedor quiero ingresar a la plataforma de Bookinder y poder registrar un libro con ayuda del scanner para el código ISBN.

Criterios de aceptación:

- **YO COMO:** Vendedor de libros.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para mi ingreso como vendedor.
- **CUANDO:** Requiera registrar un libro en la plataforma.
- **ENTONCES:** Utilizar el scanner y el ISBN para registrar el libro.

Pruebas:

- Se debe de ingresar a la plataforma Bookinder para realizar el registro de un libro por medio del scanner para código ISBN y visualizar la información del libro correctamente recuperada.

Datos importantes para la prueba: Datos del libro.

Código: P-013.

Título: Registrar usuario.

Usuarios implicados: Usuario.

Yo como usuario de la plataforma Bookinder independientemente de mi rol requiero poder registrarme en la plataforma.

Criterios de aceptación:

- **YO COMO:** Usuario de la plataforma Bookinder.
- **QUIERO:** Que la plataforma de Bookinder se encuentre habilitada para mi registro como cualquier tipo de usuario.
- **CUANDO:** Ingrese a la página de registro.
- **ENTONCES:** Permitir ingresar mis datos para el registro en la plataforma y una vez ingresados informar sobre el registro exitoso para permitir el acceso.

Pruebas:

- Se debe de realizar el ingreso a la página de registro de Bookinder y habilitar el formulario para que nuestro usuario pueda ingresar su información, posteriormente si no tiene problemas de validaciones, permitir ingresar en la plataforma y guardar su información como usuario registrado.

Datos importantes para la prueba: Nombre, Apellido, Correo, Celular, Contraseña.

Código: P-015.

Título: Visualizar usuarios registrados.

Usuarios implicados: Administrador.

Yo como administrador de la plataforma Bookinder requiero una funcionalidad que me permita visualizar los usuarios registrados en la plataforma con toda su información relacionada.

Criterios de aceptación:

- **YO COMO:** Administrador de la plataforma Bookinder.
- **QUIERO:** Que exista una funcionalidad que me permita visualizar la información de los usuarios registrados.
- **CUANDO:** Se requiera hacer revisión de los usuarios registrados.
- **ENTONCES:** Debe de existir un servicio y/o pantalla que me permita visualizar a dichos usuarios con su información relevante relacionada.

Pruebas:

- Se debe ingresar al aplicativo con rol de Administrador y realizar la búsqueda de algunos usuarios para validar que la información que se recupera es coherente y correcta comparada con la del registro de los mismos usuarios.

Datos importantes para la prueba: Lista de usuarios registrados.

Código: P-016.

Título: Ingresar y editar usuario.

Usuarios implicados: Administrador y Usuario.

Yo como administrador de la plataforma Bookinder requiero una funcionalidad que me permita editar los usuarios registrados en la plataforma con toda su información relacionada.

Yo como usuario de la plataforma Bookinder requiero una funcionalidad que me permita ingresar como usuario y visualizar toda mi información asociada

Criterios de aceptación Administrador:

- **YO COMO:** Administrador de la plataforma Bookinder.
- **QUIERO:** Que exista una funcionalidad que me permita editar la información de los usuarios registrados.
- **CUANDO:** Se requiera editar la información de los usuarios registrados.
- **ENTONCES:** Debe de existir un servicio y/o pantalla que me permita editar a dichos usuarios con su información relevante relacionada.

Criterios de aceptación Usuario:

- **YO COMO:** Usuario de la plataforma Bookinder.

- **QUIERO:** Que exista una funcionalidad que me permita ingresar mi información de registro.
- **CUANDO:** Requiera acceder a la plataforma de Bookinder.
- **ENTONCES:** Debe de existir un servicio y/o pantalla que me permita ingresar dicha información y tras el correcto ingreso de estas permita acceder a mi cuenta.

Pruebas:

- Se debe ingresar al aplicativo con rol de Administrador y realizar la búsqueda de algunos usuarios para validar que la información que se recupera es coherente y correcta comparada con la del registro de los mismos usuarios posteriormente se debe de editar la información de dichos usuarios y confirmar que la información editada se guardó correctamente.

Datos importantes para la prueba: Lista de usuarios registrados.

12. Identificación de riesgos y contingencias

Riesgos:

Datos Inconsistentes:

- **Riesgo**: La presencia de datos inconsistentes en la base de datos podría afectar las operaciones CRUD.
- **Contingencia**: Realizar una revisión exhaustiva de la consistencia de los datos antes de ejecutar las pruebas y coordinarse con los equipos de desarrollo para corregir cualquier inconsistencia identificada.

Limitaciones de Acceso a Datos de Prueba:

- **Riesgo**: La falta de acceso a datos de prueba representativos podría limitar la cobertura de las pruebas.
- **Contingencia**: Generar conjuntos de datos de prueba ficticios o utilizar herramientas para simular datos en casos donde los datos reales no estén disponibles.

Cambios en el Esquema de Datos:

- **Riesgo**: Cambios en el esquema de datos podrían afectar las operaciones CRUD.
- **Contingencia**: Coordinarse con los equipos de desarrollo para obtener información anticipada sobre cambios en el esquema y ajustar los scripts de prueba según sea necesario.

Problemas de Integración con Otros Componentes:

- **Riesgo**: La integración con otros componentes podría generar problemas que afecten las operaciones CRUD.
- **Contingencia**: Realizar pruebas de integración exhaustivas y mantener una comunicación constante con los equipos responsables de los otros componentes.

Rendimiento Insatisfactorio:

- **Riesgo**: Las operaciones CRUD pueden afectar el rendimiento, especialmente con grandes volúmenes de datos.
- **Contingencia**: Realizar pruebas de rendimiento para identificar y abordar cuellos de botella. Optimizar consultas y procesos según sea necesario.

Actualizaciones No Atómicas:

- **Riesgo**: Operaciones de actualización no atómicas podrían dejar el sistema en un estado inconsistente.
- **Contingencia**: Utilizar transacciones para garantizar que las operaciones de actualización sean atómicas y reversibles en caso de errores.

Contingencias Generales:

Documentación Incompleta:

- **Riesgo**: Falta de documentación clara sobre la API y sus endpoints.
- **Contingencia**: Colaborar con los equipos de desarrollo para obtener detalles adicionales y crear documentación de prueba detallada.

Fallos en la Seguridad:

- **Riesgo**: Vulnerabilidades de seguridad podrían exponer la API a ataques.
- **Contingencia**: Realizar pruebas de seguridad y colaborar con equipos de seguridad para abordar y corregir cualquier vulnerabilidad identificada.

Cambios No Comunicados:

- **Riesgo**: Cambios no comunicados en la API podrían afectar las pruebas planificadas.
- **Contingencia**: Mantener una comunicación constante con los equipos de desarrollo y realizar pruebas exploratorias para adaptarse a cambios inesperados.

Escasez de Recursos de Prueba:

- **Riesgo**: Recursos insuficientes para realizar pruebas exhaustivas.
- **Contingencia**: Priorizar pruebas críticas, automatizar pruebas repetitivas y buscar la asignación de recursos adicionales si es necesario.

13. Recursos y Cronograma de las pruebas

- **Herramientas de Pruebas de Carga y Seguridad:** Uso de herramientas específicas para evaluar el rendimiento y la seguridad.
- **Ingenieros Especializados:** Personal con experiencia en pruebas de rendimiento y seguridad en entornos Java y Spring Boot.
- **Entorno de Pruebas:** Configuración del entorno que refleje de manera precisa el entorno de producción.

  Preparación del Entorno:
  1. Configuración de Ambiente de Pruebas:
  2. Asegúrate de que el entorno de pruebas esté configurado correctamente, replicando el entorno de producción en la medida de lo posible.
  3. Carga de Datos de Prueba:
  4. Carga los datos necesarios para ejecutar los casos de prueba, incluyendo datos de libros, usuarios y cualquier otro elemento relevante.
- **Cronograma u orden de Ejecución:**

  1. **Pruebas Unitarias**:
     - Ejecuta las pruebas unitarias en cada componente aislado. Verifica que las funciones individuales operen correctamente y que las operaciones básicas sean exitosas.

2. **Pruebas de Funcionalidad:**
    - Ejecuta las pruebas de funcionalidad para validar que el sistema cumpla con los requisitos funcionales establecidos. Asegúrate de que las funciones clave, como el registro y búsqueda de libros, funcionen según lo esperado.
3. **Pruebas de Integración:**
    - Ejecuta las pruebas de integración para garantizar que los distintos módulos interactúen correctamente. Verifica que la información fluya sin problemas entre los componentes.
4. **Pruebas de Seguridad:**
    - Ejecuta las pruebas de seguridad para identificar y corregir posibles vulnerabilidades. Asegúrate de que el sistema resista intentos de acceso no autorizado y otros posibles ataques.

Evaluación de resultado

Registro de Resultados de Pruebas:

- Documenta los resultados de cada prueba, indicando si fue exitosa, fallida o si requiere acciones correctivas.

Registro de Errores:

- Registra cualquier error o defecto encontrado durante la ejecución de pruebas. Proporciona detalles específicos, incluyendo pasos para reproducir el error.

Análisis de Resultados:

- Analiza los resultados de las pruebas para identificar patrones, tendencias o áreas críticas que requieran atención.

Priorización de Correcciones:

- Prioriza los errores y problemas identificando en aquellos que tienen un impacto más significativo en la funcionalidad osificados, centrá seguridad del sistema.

Iteración y Retesting:

- Realiza ajustes según sea necesario y vuelve a ejecutar las pruebas afectadas para confirmar que las correcciones han sido exitosas.

Pruebas de Regresión:

- Ejecuta pruebas de regresión para asegurar que las nuevas modificaciones no hayan introducido nuevos problemas en áreas previamente funcionales.

Costos:

- **Capacitación del Personal:** Costos asociados con la formación del personal en el uso de herramientas específicas y las tecnologías involucradas.
- **Herramientas de Pruebas:** Posibles gastos relacionados con la adquisición de herramientas de pruebas de carga y seguridad.
- **Configuración del Entorno:** Gastos asociados con la configuración y mantenimiento del entorno de pruebas.

La ejecución de pruebas es un proceso iterativo y colaborativo que requiere una comunicación efectiva entre los equipos de desarrollo y de pruebas. La atención a los detalles y la documentación adecuada son clave para el éxito de esta fase.

14. Aprobaciones y Resultados

Seguimiento y reporte
Informe y Comunicación:
- Elabora un informe detallado que resuma los resultados de la ejecución de pruebas. Incluye estadísticas, tendencias y acciones tomadas.

Comunicación con el Equipo:
- Comunica los resultados a los miembros del equipo de desarrollo y otros stakeholders, destacando áreas de mejora y acciones recomendadas.

Validación de Cumplimiento:
- Asegúrate de que todos los requisitos funcionales y de seguridad hayan sido cumplidos. Verifica que el sistema esté listo para la siguiente fase del ciclo de desarrollo.

Entregables:
- **Informe de Rendimiento:** Detalles sobre tiempos de respuesta, utilización de recursos y comportamiento bajo carga.
- **Registro de Errores:** Detalles de cualquier problema identificado durante las pruebas, clasificados por gravedad y acciones correctivas tomadas.
- **Documentación de Seguridad:** Información detallada sobre las medidas de seguridad implementadas y los resultados de las pruebas de seguridad.

Resultados:
- **Informe de Evaluación de Rendimiento:** Datos cuantitativos sobre el rendimiento bajo diversas condiciones.
- **Informe de Seguridad:** Validación de las medidas de seguridad implementadas y posibles mejoras recomendadas.
- **Informe de Calidad:** Confirmación de cumplimiento de requisitos funcionales y no funcionales, así como recomendaciones para optimizar la eficiencia y seguridad.