

Implementar pruebas de software en un proyecto asegura la calidad y el correcto funcionamiento de la aplicación. Existen diferentes tipos de pruebas, entre las más importantes podemos encontrar las pruebas de carga, estas miden el rendimiento de la aplicación bajo una gran demanda. Por otro lado tenemos las pruebas funcionales que están nos ayudan a validar que la aplicación cumpla los requisitos funcionales para que la aplicación funcione correctamente, y por último tenemos las pruebas de seguridad que están como su nombre lo dice aseguran la seguridad de la aplicación identificando las posibles vulnerabilidades para así proteger la información del sistema.

Estas pruebas las podemos realizar con herramientas como Selenium y JMeter que nos permiten automatizar estos procesos, haciéndonos más fácil la ejecución de pruebas funcionales y de rendimiento. Por otra parte, se pueden implementar recursos como una matriz de trazabilidad que nos puede ayudar a relacionar estos requisitos con el tipo de prueba que se debe implementar en cada uno de ellos y una CheckList que nos asegura que estas pruebas y actividades ya han sido realizadas y completadas correctamente.

**Asignación de Roles:**

**Juan Pablo Leal Usuga:** Presentador

**Juan David Castaño Hoyos:** Equipos de pruebas

**Karen Yuliana Valencia Ochoa:** Auditor de pruebas

**Kelly Sandrith Diaz Bermudez:** Documentador

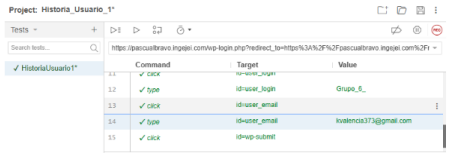
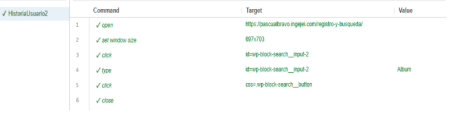
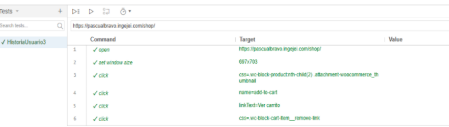
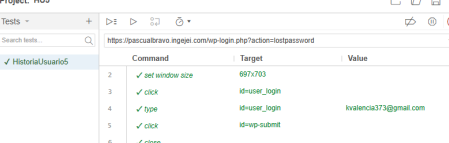
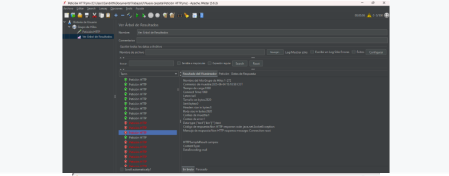
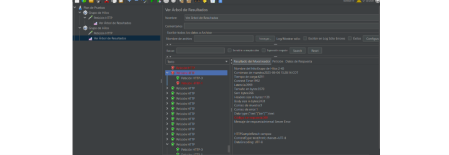
ID	Tt	Categoría	🔗	Prioridad	Tt	Fuente/Source	Tt	Objetivo	Tt	Herramienta
F1		Funcional		Urgente		<a href="https://pascualbravo.ingejei.com/wp-login.php?action=register">https://pascualbravo.ingejei.com/wp-login.php?action=register</a> (REGISTRO)		Verificar que el flujo de registro de usuario funcione correctamente, para que los nuevos usuarios puedan crear su cuenta sin errores ni validaciones faltantes.		Selenium
F2		Funcional		Moderado		<a href="https://pascualbravo.ingejei.com/registro-y-busqueda/">https://pascualbravo.ingejei.com/registro-y-busqueda/</a> (BARRA BUSQUEDA)		Validar la búsqueda de productos por palabra clave, para que los resultados sean relevantes y muestren la información completa (imagen, precio, stock).		Selenium
F3		Funcional		Urgente		<a href="https://pascualbravo.ingejei.com/cart/">https://pascualbravo.ingejei.com/cart/</a> (CARRITO)		Probar el proceso de añadir y eliminar productos del carrito, para que el total de la compra se actualice correctamente y no queden residuos de artículos.		Selenium
F4		Funcional		Urgente		<a href="https://pascualbravo.ingejei.com/checkout/">https://pascualbravo.ingejei.com/checkout/</a> (PAGINA TRANSACCIONAL)		Comprobar que el flujo de pago con tarjeta (WooCommerce) se complete sin fallos, para que el pedido se genere correctamente y se notifique al usuario.		Selenium
F5		Funcional		Critico		<a href="https://pascualbravo.ingejei.com/checkout/">https://pascualbravo.ingejei.com/checkout/</a> (PAGINA TRANSACCIONAL) <a href="https://pascualbravo.ingejei.com/wp-login.php?action=lostpassword">https://pascualbravo.ingejei.com/wp-login.php?action=lostpassword</a> (REQUERIR CONTRASEÑA)		Validar el envío de correos transaccionales (confirmación de pedido, restablecer contraseña), para que los usuarios reciban siempre la notificación adecuada.		Selenium

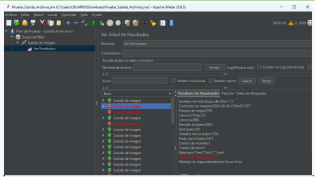
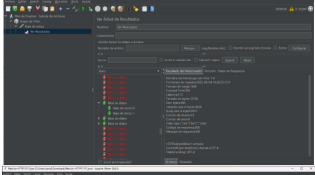
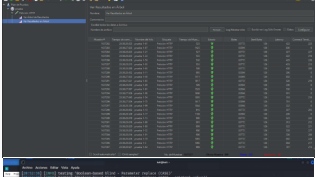
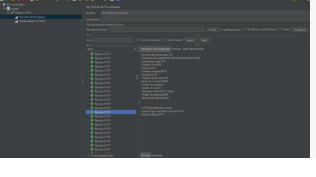
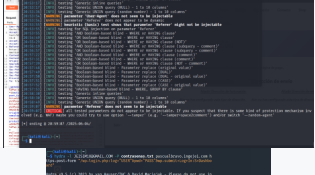
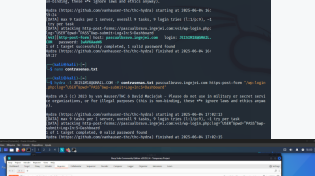


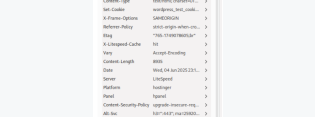

ID	Categoría	🔗	Prioridad	T↑	Fuente/Source	T↑	Objetivo	T↑	Herramienta
C1	Rendimiento		Critico		<a href="https://pascualbravo.ingejei.com/">https://pascualbravo.ingejei.com/</a> (INICIO)		Simular 500 usuarios concurrentes navegando por la página de inicio, para que el tiempo de respuesta se mantenga por debajo de 2 s bajo alta demanda.		JMeter
C2	Rendimiento		Urgente		<a href="https://pascualbravo.ingejei.com/checkout/">https://pascualbravo.ingejei.com/checkout/</a> (CHECKOUT)		Medir el tiempo de respuesta al procesar un checkout con 100 usuarios simultáneos, para que el sistema escale adecuadamente sin errores de timeout.		JMeter
C3	Rendimiento		Moderado		<a href="https://pascualbravo.ingejei.com/wp-admin/post-new.php?post_type=product">https://pascualbravo.ingejei.com/wp-admin/post-new.php?post_type=product</a> (AÑADIR PRODUCTOS)		Ejecutar un test de estrés subiendo archivos grandes (imágenes de producto) en paralelo, para que la aplicación soporte cargas masivas sin caídas.		JMeter
C4	Rendimiento		Moderado		<a href="https://pascualbravo.ingejei.com/wp-admin/admin.php?page=litespeed-db_optm">https://pascualbravo.ingejei.com/wp-admin/admin.php?page=litespeed-db_optm</a> (BASE DE DATOS)		Analizar el rendimiento de la base de datos bajo 200 consultas/segundo, para que no haya cuellos de botella en la capa de datos.		JMeter
C5	Rendimiento		Urgente		<a href="https://pascualbravo.ingejei.com/">https://pascualbravo.ingejei.com/</a> (INICIO)		Realizar un test de resistencia continuo durante 2 horas con 100 usuarios, para detectar fugas de memoria o degradación progresiva del servicio.		JMeter

ID	Tt	Categoría	🔗	Prioridad	Tt	Fuente/Source	Tt	Objetivo	Tt	Herramienta
S1		Seguridad		Critico		<a href="https://pascualbravo.ingejei.com/my-account/">https://pascualbravo.ingejei.com/my-account/</a> (Inicio de sesion)		Ejecutar escaneos de vulnerabilidades (SQL Injection) en todos los formularios de entrada, para que la aplicación esté protegida contra inyecciones maliciosas.		Burpsuite, sqlmap
S2		Seguridad		Urgente		<a href="https://pascualbravo.ingejei.com/my-account/">https://pascualbravo.ingejei.com/my-account/</a> (Inicio de sesion)		Probar la fuerza bruta de inicio de sesión con un diccionario de contraseñas, para que validar que los mecanismos de bloqueo de cuenta y captcha funcionan.		Hydra
S3		Seguridad		Moderado		<a href="https://pascualbravo.ingejei.com/wp-login.php">https://pascualbravo.ingejei.com/wp-login.php</a> (Inicio de sesion Admin)		Auditar la gestión de permisos (roles de usuario) intentando accesos no autorizados, para que sólo los roles adecuados puedan ver o modificar información sensible.		Burp Suite
S4		Seguridad		Critico		<a href="https://pascualbravo.ingejei.com/">https://pascualbravo.ingejei.com/</a> (Inicio)		Revisar los encabezados HTTP de seguridad (CSP, HSTS, X-Frame-Options), para que esté mitigada la mayoría de ataques de inyección y clickjacking.		Burp Suite
S5		Seguridad		Urgente		<a href="https://pascualbravo.ingejei.com/my-account/">https://pascualbravo.ingejei.com/my-account/</a> (Inicio de sesion)		Probar la protección contra CSRF enviando formularios y peticiones con y sin el token CSRF válido, para que se evite que atacantes realicen acciones no autorizadas en nombre de un usuario autenticado.		Burp Suite

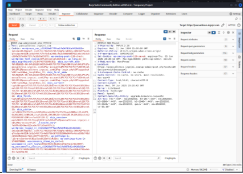
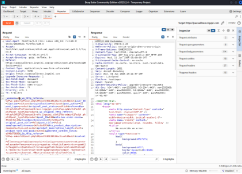
ITEM	CATEGORÍA	CHECKLIST (Qué se verifica?)	¿Cómo se verifica?	VERIFICACION
A1	Inyección	¿Se puede acceder a los datos con seguridad? ¿Están bien definidos los roles ?	Ingresar desde la ruta de login desde la aplicación y validar que esté activo el https desde el navegador y después de ingresar validar el network (consola de desarrollador) u otra herramienta que permita capturar la transmisión de datos (SELENIUM). Verificar desde la cuenta del administrador de la aplicación si se puede crear usuarios con diferentes roles y autenticarse luego. (posteriormente pasar la verificación número 1)	Ingresar al login de la pagina, validar que la URL utiliza el protocolo https, iniciar sesión y comprobar que todas las solicitudes sigan utilizando https, asegurandose que no hayan filtraciones de credenciales.
A2	Pérdida de autenticación y gestión de sesiones	¿Está habilitada la autenticación en dos pasos? ¿Se configuran correctamente las llaves secretas en WordPress?	Validar que las sesiones de los usuarios expiren tras un tiempo adecuado de inactividad. Probar si está habilitado el doble factor de autenticación iniciando sesión desde otro navegador/dispositivo. Revisar el archivo wp-config.php y comprobar que las llaves secretas no están vacías. Esto se podría probar con selenium.	Iniciar sesión en la pagina como usuario común, dejar la sesión inactiva durante un tiempo determinado (20min - 45min), recargar la pagina, el sistema redirige al login e indique que la sesión a expirado. Iniciar sesion desde otro navegador, se verifica si envia codigo de autenticación.
A3	Datos sensibles accesibles	¿Se gestionan correctamente los permisos de acceso a datos sensibles? ¿Se protege la base de datos frente a accesos externos?	Probar que un usuario sin permisos no puede ver información sensible (como datos de tarjetas o direcciones personales) que hayan en la aplicación. Revisar que haya reglas en el firewall que bloqueen el acceso al puerto de la base de datos desde direcciones IP públicas. Esto se verificara por medio de burp suite.	Iniciar sesion como usuario común sin privilegios intentar acceder a paneles administrativos, datos financieros, interceptar las respuestas http con burp suite aseurando que estos datos sensibles no esten visibles.
A4	Entidad externa de XML (XXE)	¿La aplicación procesa XML de forma segura? ¿Se han deshabilitado las entidades externas en los parsers XML?	Verificar en el código fuente que no se estén utilizando parsers XML inseguros como DOM, XMLReader, XMLWriter sin configuración de seguridad. Realizar pruebas enviando payloads XXE maliciosos y verificar que no se procesan. Revisar dependencias para asegurar que no incluyen parsers XML vulnerables. Se va a verificar mediante burp suite y OWASP	Localizar archivos ue procesan XML, buscar parses que por defecto aceptan entidades externas, revisar que esten deshabilitados entidades externas, reemplazar por parsers más seguros.
A5	Control de acceso inseguro	¿Existen copias de seguridad seguras y accesibles? ¿Se validan las llamadas a las APIs para evitar accesos no autorizados?	Inspeccionar las llamadas API para detectar accesos no autorizados. Confirmar que existan copias de seguridad funcionales y que estén almacenadas en un lugar seguro. Se verificara mediante burp suite.	Interceptar con burp suite las llamadas API desde el navegador, identificar las ruas expuestas. Revisar el control de acceso accediendo a rutas sensibles, asegurar que requieran autenticación. Comprobar archivos de respaldo y asi poder hacer una restauración de prueba en un ambiente controlado, verificando que todos lo archivos carguen correctamente.

A6	Configuración de seguridad incorrecta	¿Se han cambiado todas las configuraciones por defecto? ¿Los permisos de archivos y carpetas son los adecuados? ¿Se han asegurado archivos críticos como .htaccess y wp-config.php?	Verificar que no se usan credenciales, puertos o configuraciones por defecto, por ejemplo admin/admin. Revisar permisos en el servidor: htaccess y wp-config.php: 400 o 440. Confirmar que la autenticación y la gestión de sesiones están bien configurados en la aplicación. Escanear servicios expuestos innecesariamente. Se verificara a traves de Hydra.	Se prueban los usuarios comunes con contraseñas por defecto revisando el archivo wp-config.php. Revisar con ls -l asegurando permisos como 400 o 440 que esten debidamente protegidos con directivas apache. Escanear puertos abiertos y protgerlos.
A7	Cross site scripting (XSS)	¿Se impide la ejecución de scripts en comentarios, formularios u otros campos?	Implementar una política de seguridad CSP en el navegador que bloquee scripts no autorizados. Se puede verificar a traves de google CSP evaluator.	Se abre el navegador la pagina protegida, se realiza el escaneo de las cabeceras http, se intenta poner un script malicioso desde la consola, este deberia ser bloqueado.
A8	Decodificación insegura	¿Se evita deserializar datos provenientes de usuarios o fuentes externas no confiables? ¿Se validan los datos antes de deserializarlos?	Verificar que se validan los datos con firmas digitales antes de deserializar. Revisar el código en busca de funciones como unserialize(), json_decode(), yaml_parse() y comprobar si los datos provienen de formularios, cookies, APIs, etc. Se verificara por medio de burp suite.	Se verifica el uso de funciones inseguras asegurandose de donde vienen los datos, es decir el origen, que pueden ser de usuarios, formularios. Revisar parámetros GET/POST.
A9	Componentes con vulnerabilidades	¿Se mantienen actualizados todos los componentes del sistema? ¿Se evitan servicios de hosting inseguros? ¿Se han eliminado dependencias obsoletas o sin soporte?	Revisar los componentes como plugins o liberias <b>usando herramientas como OWASP</b> . Verificar en las bases de datos de vulnerabilidades, si alguno de los componentes usados tiene fallos conocidos. Comprobar que el hosting cumple con estándares de seguridad. Eliminar o actualizar software obsoleto mediante auditorías periódicas.	Se revisa las versiones especificas de los componentes de las bases de datos, Revisar alertas de seguridad. Comprobar la configuración SSL/TLS y politicas de seguridad HTTP. Realizar un debido control de versiones.
A10	Insuficiente monitorización y registro	¿Existe un sistema completo de logging? ¿Se monitorizan actividades sospechosas? ¿Se cumplen con los requisitos RGPD?	Verificar la existencia de logs detallados (accesos, errores, cambios). Buscar código peligroso mediante análisis estático. Configurar alertas para actividades inusuales. Comprobar que los plugins RGPD registran adecuadamente el tratamiento de datos.	Se valida los logs que esten activos y no vacíos, se confirma que registran fecha, hora, IP, acción realizada. Se verifica que esten debidamente protegidos contra accesos no autorizados.

ID	Tt	Descripción	Tr	Precondición	Tr	Entrada	Responsable pruebas	Resultado Esperado	Resultado Obtenido	Resultado	Resultado 2
F1	Registro de Usuario	La página de registro está disponible		Correo, contraseña	Juan David castaño	El sistema permite que un nuevo usuario se registre correctamente, creando cuenta sin errores.	Al realizar la prueba se valida correctamente los datos ingresados y se genera correctamente la cuenta y aparece un mensaje de confirmación de cuenta nueva creada.				
F2	Busqueda de Productos	El usuario debe de haber iniciado sesión	Buscar "Palabra clave" en la barra de búsqueda	Juan David castaño	El sistema permite la búsqueda de productos por palabra clave posterior a esto muestra justo la información completa del producto	Al realizar la prueba se puede evidenciar que funciona correctamente la funcionalidad de la barra de búsqueda y trae exitosamente la información completa del producto (imagen, precio, stock)					
F3	Verificar el Carrito	El usuario debe de haber iniciado sesión	Ingresar y eliminar artículos al carro	Juan David castaño	Se espera que el usuario pueda añadir y eliminar productos del carrito sin ningún problema aparte de eso que se actualice correctamente el carrito.	Al probar esta funcionalidad el sistema permitió añadir y eliminar productos del carrito correctamente, el total de la compra se actualizó en tiempo real y no quedaron residuos.					
F4	Pasarela de Pagos	El usuario debe de haber iniciado sesión y tener artículos en el carrito	Elegir un metodo de pago	Juan David castaño	El usuario completa exitosamente el proceso de pago con tarjeta, el pedido se genera correctamente en el sistema de woocommerce y posterior a esto que el usuario reciba una confirmación del pedido tanto en pantalla como al correo electrónico.	Al verificar esta funcionalidad se halla un error al no encontrar un metodo de pago en el sistema WooCommerce, esta funcionalidad no esta activada y al no poder acceder a esto, tampoco puede realizar pedido, ni confirmación de correo. Revisar funcionalidad.	<div>No se ha facilitado ningún método de pago.</div> <div>Opciones de pago</div> <div>No hay ningún método de pago disponible. Esto puede ser error nuestro. Por favor, contactanos si necesitas ayuda para realizar tu pedido.</div>				
F5	Notificacion de Correo	Tener diligenciado y actualizado el correo electrónico.	Dar click boton "enviar correo electronico"	Juan David castaño	El sistema envíe los correos electrónicos esperados (Confirmación de pedido, Restablecer contraseña) y los usuarios reciban la notificación sin problemas.	Al realizar la prueba se puede evidenciar que el envío de correos trasaccionales para confirmación de pedido no funciona ya que el WooCommerce esta defectuoso, por otro lado, las notificaciones para restablecer contraseña están en correcto funcionamiento.					
F6	Tiempo de respuesta	Pagina de inicio disponible	500 hilos simulados.	Juan David castaño	El sistema debe ser capaz de responder a los 500 usuarios concurrentes en el tiempo de respuesta de 2 segundos, sin que se caiga la pagina.	Probando esta historia de usuario se encontro que al alcanzar apenas los 269 usuarios la pagina empieza a tener dificultades de rendimiento ya que al parecer es mucha carga en tan poco tiempo, se cae la pagina por un tiempo corto, por lo que se recomienda una revisión.					
F7	Procesamiento Check Out	Inicio de sesión, carrito con productos.	100 hilos simulados.	Juan David castaño	Se espera que los 100 usuarios simultaneos completen exitosamente el proceso de checkout sin dificultades.	Durante la prueba de carga con 100 usuarios realizando el proceso del checkout , se mostraron varios errores 500 (Internal server error) significa que algo esta mal en el servidor del sitio web y que el sistema no pudo manejar la concurrencia esperada.					

ID	Tt	Descripción	Tt	Precondición	Tt	Entrada	Responsable pruebas	Resultado Esperado	Resultado Obtenido	Resultado	Resultado 2
F8	Soporte Cargas Masivas	Archivos activos, sistema preparado para aceptar imagenes grandes.	Varios hilos simultaneamente y subir archivos grandes.	Juan David castaño	Se espera que al subir archivos grandes el sistema soporte cargas masivas sin caidas.	Al realizar la prueba de estres con subida de archivos grandes en paralelo, se observó que el sistema logró procesar una parte significativa, pero tambien se presentaron errores de tiempo de espera, lo que indica que el sistema no es capaz de mantener las cargas simultaneas.					
F9	Cuello Botella	El sistema debe estar desplegado y funcionando correctamente en un entorno estable, accesible para los usuarios de prueba.	No se requieren datos especificos de entrada	Juan David castaño	El sistema debe responder correctamente al realizar las 200 consultas por segundo sin afectar la pagina.	Durante la prueba con 200 consultas por segundo, se observaron inestabilidades en el tiempo de respuesta y algunos errores, el rendimiento es aceptable pero no el óptimo.					
F10	Degradacion de Servicios	El sistema debe estar desplegado y funcionando correctamente en un entorno estable, accesible para los usuarios de prueba.	No se requieren datos especificos de entrada	Juan David castaño	El sistema debe soportar la carga continua de 100 usuarios durante 2 horas sin afectar el rendimiento, sin caidas, ni errores.	Al realizar la prueba se observo que el sistema se mantuvo estable y consistente durante el tiempo definido de 2 horas con 100 usuarios, no hubo caidas, ni errores, no se detecto indicios de fuga de memoria o degradación progresiva del servicio.		 			
F11	Vulnerabilidades SQL	La pagina debe de estar contar con un entorno de pruebas como Kali y la herramienta sqlmap	Archivo datos.txt con la petición HTTP POST al formulario de login, incluyendo campos como log, pwd, y cabeceras. El archivo se utilizó como entrada para sqlmap con parámetros --level=5 --risk=3 para realizar el escaneo de inyecciones SQL.	Juan David castaño	No detectar vulnerabilidades de inyección SQL en ningún formulario, confirmando que las entradas están correctamente validadas.	No se encontraron vulnerabilidades de inyección SQL en ninguno de los formularios analizados.					
F12	Inicio de Sesion	Tener diccionario de contraseñas y login accesible.	Usar el comando: hydra -l JEISIM18@GMAIL.COM -P contraseñas.txt pascalbravo.ingejel.com https-post-form /wp-login.php:log=USER&pwd=PASS&wp-submit=Log+S=Dashboard y el archivo contraseñas.txt que contiene múltiples contraseñas, incluida la correcta.	Juan David castaño	Los mecanismos de seguridad de la pagina bloquean o limiten los intentos tras varios accesos fallidos, evitando el acceso no autorizado incluso si la contraseña correcta está en el ataque.	Se logró acceder correctamente cuando la contraseña correcta estaba dentro del diccionario probado con Hydra, lo que indica ausencia o insuficiencia de mecanismos de protección contra fuerza bruta.		 			
F13	Gestion de Permisos	Tener varias cuentas creadas con diferente rol.	Credenciales de usuario sin permisos. Solicitudes HTTP dirigidas a recursos administrativos especificos.	Juan David castaño	Los usuarios sin los roles adecuados no puedan acceder ni modificar información sensible, asegurando control efectivo de permisos.	No se permitió acceso a usuarios sin los permisos correspondientes, confirmando que la gestión de roles está correctamente implementada.					
F14	Seguridad HTTP	Pagina accesible con https activado.	Solicitud HTTP GET a la página principal. Analisis de headers de respuesta del servidor	Juan David castaño	La aplicación implemente correctamente encabezados de seguridad HTTP para protegerse contra inyección de código y clickjacking.	Se encontró el encabezado CSP, pero con configuración básica y sin restricciones a fuentes externas. HSTS está ausente, exponiendo a posibles ataques MITM. X-Frame-Options está correctamente configurado como SAMEORIGIN. Encabezados como X-XSS-Protection y X-Content-Type-Options no están implementados.		 			



ID	T+	Descripción	T+	Precondición	T+	Entrada	Responsable pruebas	Resultado Esperado	Resultado Obtenido	Resultado	Resultado 2
F15		Proteccion CSRF		Tener formularios o acciones protegidas.		Petición HTTP con token CSRF válido incluido en el formulario/header Petición HTTP idéntica pero sin token CSRF	Juan David castaño	Las peticiones sin token CSRF válido sean rechazadas o redirigidas, asegurando que solo acciones legítimas del usuario autenticado sean aceptadas.	Al enviar formularios sin el token CSRF válido, la aplicación redirige al usuario a la página de inicio de sesión, lo que indica que la protección contra CSRF está activa y funcional.		

PRUEBAS	CONCLUSIONES	RECOMENDACIONES
FUNCIONALES	<p>Con la realización de las pruebas funcionales permitieron corroborar el correcto ejecución de la mayoría de las funcionalidades del sistema. Se verificó que el gestor de usuario, el envío de correos de recuperación, la búsqueda de productos y el carrito actúan correctamente.</p> <p>Sin embargo, se evidenció que el gestor de pagos de WooCommerce no permitió el pago por medio de tarjetas de crédito. Por lo tanto, tampoco envía las confirmaciones de pago. Se cree que esto sucede porque esta funcionalidad no está habilitada o correctamente configurada.</p>	<p>-Revisar que WooCommerce esté correctamente configurado, además de asegurarse que su funcionamiento sea correcto, permitiendo el pago con tarjetas y otros medios de pago.</p> <p>-Realizar nuevamente las pruebas de envío de notificaciones, verificando que el correo de confirmación de pagos se envíe.</p>
RENDIMIENTO	<p>Las pruebas de rendimiento nos mostraron que la página responde de manera adecuada cuando el flujo simultaneo no supera los 100 usuarios navegando por el sistema. Por lo tanto el sistema se mantiene estable sin registrar ninguna novedad ni caídas del servidor.</p> <p>No obstante, al aumentar la exigencia al servidor, con un flujo masivo de usuarios concurrentes, podemos notar fallos. Específicamente, se identificaron caídas parciales del servidor cuando se realizaron inserciones masivas en la base de datos o se subieron archivos de gran tamaño. Por lo tanto, el sistema no está preparado para ser usado a gran escala, su uso por el momento debe ser de baja exigencia.</p>	<p>-Se recomienda usar en entornos de baja exigencia mientras se realizan las mejoras de rendimiento.</p> <p>-Optimizar las consultas y la insercción de datos que se realizan a la base de datos.</p> <p>-Implementar un sistema de colas para gestionar cargas elevadas de procesos como la subida de archivos.</p>
SEGURIDAD	<p>"Las pruebas de seguridad muestran que la aplicación tiene una adecuada protección contra inyección SQL, gestión correcta de permisos y roles, implementación efectiva de encabezados HTTP de seguridad y protección CSRF funcional. Sin embargo, se detectó una vulnerabilidad crítica en el sistema de autenticación: la ausencia de protección contra ataques de fuerza bruta permitió el acceso exitoso mediante un ataque de diccionario, evidenciando la falta de mecanismos como bloqueo temporal de cuenta, limitación de intentos o CAPTCHA."</p>	<p>-Incorporar un sistema de bloqueo temporal tras varios intentos fallidos consecutivos al iniciar sesion.</p> <p>-Añadir CAPTCHA o reCAPTCHA en el formulario de inicio de sesión para diferenciar usuarios legítimos de bots.</p> <p>-Considerar alertas de seguridad y monitoreo para detectar patrones sospechosos de intentos de acceso.</p>

#### CONCLUSIONES GENERALES

El sistema mostró que sus funcionalidades se comportan y responden adecuadamente, mostró fallas al procesar sus pagos pero con una correcta configuración de la implementación se puede solucionar. En cuanto a su rendimiento, el sistema es estable cuando trabaja a menor escala, no es adecuado para trabajarse en entornos de mucha exigencia pues presenta caídas de rendimiento. En temas de seguridad se mostró seguro en el momento de realizar las inyecciones, el protocolo HTTP y el uso del CSRF permiten mejorar su seguridad. Sin embargo, sus vulnerabilidades en el sistema de autenticación mostraron que está expuesta a ataques externos, esto debe ser solucionado antes de ponerse en funcionamiento. En conclusión, el sistema no está preparado aún para un uso a gran escala ni expuesto a entornos de alta exigencia, por esto, se recomienda ser usado en contextos de baja demanda mientras se corrigen sus fallas y se refuerza su seguridad.