ITEM	CATEGORÍA	CHECKLIST (Qué se verifica?)	¿Cómo se verifica?
A1	*Inyección*	**¿Se puede acceder a los datos con seguridad? ¿Están bien definidos los roles?**	1. Ingresar desde la ruta de login de la aplicación y validar que esté activo el protocolo HTTPS desde el navegador. Después de ingresar, validar el network (consola de desarrollador) u otra herramienta que permita capturar la transmisión de datos (Selenium). 2. Verificar desde la cuenta del administrador de la aplicación si se pueden crear usuarios con diferentes roles y autenticarse luego (posteriormente pasar la verificación número 1).
A2	*Pérdida de autenticación y gestión de sesiones*	**¿Se puede acceder a secciones protegidas sin iniciar sesión? ¿Las sesiones expiran de manera adecuada?**	1. Inicia sesión desde el login y accede a alguna sección de la página que solo sea accesible si has iniciado sesión (por ejemplo: tu carrito de compras, el checkout de un producto o el panel de administrador para agregar productos). Luego, copia la URL de esa página y cierra sesión. Después, pega esa URL en el navegador para verificar si puedes acceder sin haber iniciado sesión o si te redirige al login u otra sección que impida el acceso. Esta prueba se puede hacer de forma manual o utilizando Selenium. 2. Iniciar sesión normalmente en la aplicación y permanecer inactivo por un tiempo (10 a 30 minutos). Verificar que el sistema detecta la inactividad y que, al momento de ejecutar una acción que requiera una sesión activa o que esté restringida por sesión (como entrar a mi perfil, preferencias, etc.), se redirija efectivamente a una página de login o que indique un mensaje de sesión terminada. Se puede realizar o automatizar con Selenium.
A3	*Datos sensibles accesibles*	**¿Los datos que se mandan en los formularios se mandan de manera segura y encriptada? ¿Los datos de un usuario pueden ser vistos por otros usuarios que no deberían?**	1. Verificar el tipo de protocolo de la página, viendo que esta tenga protocolo de encriptación de datos HTTPS y que estos no se envíen en texto plano, para que nadie pueda ver los datos enviados. También se puede usar BurpSuite para ver la forma en que se envían los datos y asegurarse de que estos vayan cifrados. 2. Primero, inicia sesión con una cuenta de usuario y entra a la sección donde aparezca información sensible, como los

			datos de la cuenta. Copia la URL de esa página y luego cierra sesión. Después, inicia sesión con otra cuenta diferente, pega la URL que copiaste y fíjate si te permite ver la información del primer usuario o si te redirige para que no puedas acceder a esos datos.
A4	*Entidad externa de XML (XXE)*	**¿Se permite la carga de XML en algún formulario de la aplicación?**	1. Revisar en todos los formularios de la aplicación si en alguno de ellos se necesita la carga de archivos XML. En aquellos que utilicen cargas XML, ejecutar cargas (payloads) modificadas para intentar inyectar código, recursos o estructuras de documento maliciosas.
A5	*Control de acceso inseguro*	**Se prohíbe correctamente el acceso a recursos o funcionalidades según el rol o nivel de acceso del usuario?** **¿Se previene la escalación de privilegios?**	1. Iniciar sesión con un usuario con un rol o nivel de acceso bajo (por ejemplo, un cliente) e intentar acceder a URLs o funcionalidades que requieren un rol o nivel de acceso superior. Repetir el paso anterior, pero con un rol o nivel de acceso alto (por ejemplo, un administrador). 2. Enviar con herramientas como Selenium o Postman, peticiones y llamadas a Endpoints o APIs para comprobar sus respuestas HTTP (403, 401, etc.), según las claves de acceso o la falta de estas.
A6	*Configuración de seguridad incorrecta*	**¿Existen headers de seguridad HTTP faltantes o una mala configuración (CORS, CSP, X-Frame-Options, etc.)?** **¿Hay directorios expuestos que no deberían ser públicos?**	1. Utilizar herramientas online como securityheaders.com o Mozilla HTTP Observer para escanear los headers de la página web. 2. Asegurarse de que en la configuración de Apache/NGINX los directorios y puertos expuestos sean los mínimos y necesarios. También se pueden utilizar escáneres de vulnerabilidades como OWASP ZAP o Nikto para verificar endpoints o directorios expuestos.
A7	*Cross site scripting (XSS)*	**¿Existen entradas donde se puedan inyectar scripts o código malicioso?**	1. Inyectar cargas (payloads) que alerten sobre la ejecución inadecuada de código (por ejemplo, <script>alert(1)</script>) en los campos de entrada y endpoints, es decir, búsquedas, comentarios, formularios, etc., y observar las respuestas en el navegador o en BurpSuite. 2. Verificar que el contenido ingresado por el usuario sea escapado, sanitizado y limpiado correctamente para evitar la ejecución inesperada de código.

A8	*Decodificación insegura*	**¿Se decodifican los datos o cargas externas de forma insegura (JSON, XML, etc.)?**	1. Encontrar los endpoints que reciben datos codificados (JSON, XML, etc.) y enviar cargas (payloads) modificadas que intenten ejecutar, acceder o mostrar recursos sensibles que no deberían estar expuestos al momento de decodificar estas cargas. 2. Verificar con Burp Intruder la decodificación con diferentes cargas y formatos para asegurarse de que la decodificación se ejecuta de manera segura.
A9	*Componentes con vulnerabilidades*	**¿Se usan librerías, componentes o recursos externos desactualizados y obsoletos?**	1. Verificar la lista de dependencias y plugins para confirmar que las versiones instaladas (chequeando package.json y wp plugin list) son las últimas o no están obsoletas. Estas verificaciones de vulnerabilidades se pueden realizar mediante herramientas como WPScan y Patchstack, además de consultar las bases de datos públicas de vulnerabilidades CVE.
A10	*Insuficiente monitorización y registro*	**¿Se registra de forma correcta la actividad o el tráfico sospechoso, o eventos de seguridad importantes?** **¿Los logs están activados, guardados y protegidos correctamente?**	1. Realizar acciones de seguridad importantes, como logins exitosos y fallidos, cambios de contraseñas, cambios de correo, eliminación de datos, etc., y comprobar que el log posea la información correspondiente, documentando la fecha y hora, la cuenta e IP, la acción y el resultado como información adicional. Esta información específica se obtiene codificando el logger propiamente, o a través de herramientas como Wordfence o los logs del panel de control. Se puede acceder a los logs básicos técnicos activando el modo DEBUG de WordPress. 2. Verificar que la configuración de la herramienta o código logger es correcta y que evita que los logs sean eliminados accidentalmente, o que solo sean eliminados después de mucho tiempo (se recomienda que esté configurado a que permanezcan mínimo 90 días) además, que solo se pueda acceder a ellos por usuarios o cuentas con roles o niveles de acceso correctos.