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 podria probrar 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 redirije 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 **usando herramientas como OWASP.** 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.