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)	
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.	
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.	
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.	
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.	
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.	
A7	Cross site scripting (XSS)	¿Se impide la ejecución de scripts en comentarios, formularios u otros campos?	Implementar una política de seguridad en el navegador que bloquee scripts no autorizados	
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.	
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.	
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.	