Protocolo de Seguridad en el Desarrollo
Proyecto MVP – EduTech IA

1. Objetivos del Protocolo

Integrar la seguridad en todas las fases del desarrollo (Security by Design).
Minimizar riesgos mediante la identificación y mitigación temprana de vulnerabilidades.
Asegurar la confidencialidad, integridad y disponibilidad de la aplicación y de los datos de los usuarios.
Cumplir con normativas y estándares de seguridad reconocidos.

2. Medidas y Buenas Prácticas por Fase

a) Requerimientos y Análisis de Riesgos

Definición de Requerimientos de Seguridad: Incluir requisitos específicos de seguridad desde la fase de recopilación de requerimientos.
Análisis de Amenazas y Riesgos: Realizar un modelado de amenazas (por ejemplo, utilizando OWASP Threat Modeling) para identificar posibles vectores de ataque y definir contramedidas.

b) Diseño y Arquitectura

Diseño Seguro: Aplicar principios de “Security by Design” y “Defense in Depth”.
Segregación de Entornos: Mantener entornos separados para desarrollo, pruebas y producción, minimizando la exposición de datos sensibles.
Revisión del Diseño: Validar que el diseño arquitectónico contemple mecanismos de autenticación, autorización, cifrado y manejo seguro de errores.

c) Desarrollo e Implementación

Codificación Segura: Adoptar las prácticas de codificación segura (OWASP Secure Coding Practices) y utilizar patrones de diseño que prevengan vulnerabilidades comunes.
Validación y Sanitización de Entradas: Implementar controles estrictos para validar y sanitizar todas las entradas provenientes del usuario.
Gestión de Dependencias: Mantener actualizadas las librerías y frameworks, y utilizar herramientas para el análisis de vulnerabilidades en dependencias.
Revisión de Código y Análisis Estático: Realizar code reviews periódicos y utilizar herramientas de análisis estático (como SonarQube) para identificar posibles fallos de seguridad.

d) Pruebas y Validación

Pruebas de Seguridad: Ejecutar pruebas de penetración y escaneos automatizados para detectar vulnerabilidades en la aplicación.

Integración en el Pipeline CI/CD: Incluir pruebas de seguridad en el flujo de integración y entrega continua para garantizar la detección temprana de incidencias.
Análisis Dinámico: Complementar el análisis estático con pruebas de seguridad en tiempo de ejecución.

e) Despliegue y Operación

Configuración Segura: Aplicar configuraciones seguras en servidores, contenedores y servicios, siguiendo el principio de “Least Privilege” y modelos de “Zero Trust”.
Contenerización Segura: Utilizar imágenes Docker mínimas, escanearlas regularmente y gestionar vulnerabilidades a través de registros de contenedores.
Gestión de Accesos y Monitoreo: Implementar controles de acceso robustos, autenticación multifactor y sistemas de monitoreo y logging para la detección temprana de incidentes.

f) Mantenimiento y Actualización

Actualizaciones y Parches: Realizar actualizaciones periódicas y aplicar parches de seguridad tan pronto se identifiquen vulnerabilidades.
Auditorías y Revisiones de Seguridad: Programar auditorías de seguridad internas y externas para evaluar la efectividad de las medidas implementadas.
Plan de Respuesta a Incidentes: Establecer procedimientos claros para la detección, contención y resolución de incidentes de seguridad.