1. Definición de arquitectura modular

Análisis e Informe

Una arquitectura modular divide la aplicación en componentes independientes, cada uno con una responsabilidad única y la capacidad de comunicarse a través de interfaces bien definidas. Esta estructura permite que los módulos sean intercambiables, escalables y fáciles de mantener. En ciberseguridad, la modularidad mejora la identificación y corrección de vulnerabilidades, ya que una brecha en un módulo tiene impacto limitado sobre el resto del sistema. Además, facilita el trabajo en equipo, ya que cada integrante puede desarrollar o mejorar un módulo específico de manera aislada.

Ventajas clave:

Escalabilidad y flexibilidad: Permite añadir o modificar funcionalidades sin afectar al resto del sistema.

Mejora la seguridad: El aislamiento de módulos limita el alcance de posibles brechas.

Mantenibilidad y reutilización: Los módulos pueden ser actualizados o reutilizados en otros proyectos.

Agilidad en el desarrollo: Facilita la colaboración y la integración continua.

Plan de Ejecución

Definir los módulos principales del sistema: análisis web, análisis cloud, “toquen de antesala”, interfaz, informes.

Establecer interfaces claras para la comunicación entre módulos.

Asignar responsables para cada módulo.

Documentar la arquitectura y los flujos de datos entre módulos.

Checklist

Identificar y documentar los módulos funcionales.

Definir interfaces y protocolos de comunicación.

Asignar responsables de desarrollo para cada módulo.

Redactar un documento de arquitectura modular.

Validar la arquitectura con el equipo.

2. Selección de herramientas a integrar

Análisis e Informe

La selección debe priorizar herramientas open source y ampliamente reconocidas para análisis de vulnerabilidades en web y cloud. Ejemplos recomendados:

OWASP ZAP: Escaneo de aplicaciones web, detección de XSS, SQLi y otras vulnerabilidades.

OpenVAS: Escaneo de vulnerabilidades en sistemas y redes.

Wapiti: Análisis web ligero y automatizado.

Nessus (versión gratuita o educativa): Escaneo avanzado, integración cloud y cumplimiento normativo.

ClamAV: Análisis antivirus para el “toquen de antesala”.

La integración de estas herramientas permite cubrir un espectro amplio de amenazas, automatizar pruebas y generar informes útiles para la mitigación de riesgos.

Plan de Ejecución

Investigar y comparar las funcionalidades de las herramientas candidatas.

Seleccionar las herramientas que mejor se adapten a los objetivos y recursos del equipo.

Probar la integración básica de cada herramienta en el entorno de desarrollo.

Documentar los procedimientos de integración y uso.

Checklist

Listar y comparar herramientas candidatas.

Seleccionar herramientas definitivas para web, cloud y análisis de archivos.

Realizar pruebas de integración en entorno controlado.

Documentar la integración y configuración de cada herramienta.

Validar la cobertura de amenazas y funcionalidades.

3. Diseño de la interfaz y flujo de usuario

Análisis e Informe

El diseño debe centrarse en la experiencia de usuario (UX), priorizando la claridad, accesibilidad y facilidad de uso. Es fundamental que los usuarios comprendan el estado de la seguridad de sus sistemas y reciban recomendaciones claras y contextualizadas. Se recomienda:

Dashboard centralizado: Visualización rápida de resultados y alertas.

Microinteracciones educativas: Mensajes breves que expliquen riesgos y buenas prácticas.

Accesibilidad: Cumplir con estándares para usuarios con discapacidades.

Feedback positivo: Confirmar acciones seguras y reforzar comportamientos deseados.

El flujo de usuario debe ser simple: registro/login, selección de análisis, carga de archivos, visualización de resultados e informes de mitigación.

Plan de Ejecución

Crear wireframes y prototipos de la interfaz.

Definir el flujo de usuario desde el inicio hasta la obtención de resultados.

Recoger feedback del equipo y usuarios potenciales.

Implementar la interfaz siguiendo buenas prácticas de UX y accesibilidad.

Checklist

Diseñar wireframes del dashboard y pantallas principales.

Definir el flujo de usuario y puntos de interacción.

Incorporar mensajes educativos y feedback positivo.

Validar accesibilidad y usabilidad.

Recoger y aplicar feedback iterativo.

Resumen de Mejoras Propuestas

Modularidad estricta para facilitar la escalabilidad y el mantenimiento.

Integración de herramientas líderes open source y comerciales (si es viable).

UX centrada en la educación y la accesibilidad, con tutoriales y mensajes contextuales.

Documentación exhaustiva de arquitectura, integración y diseño.