**Sección 1 .**

**Asignación de Valores de Probabilidad e Impacto en la Clasificación de Riesgos**

La evaluación y clasificación de riesgos en este proyecto se fundamenta en dos dimensiones clave: la probabilidad de ocurrencia y el impacto potencial. Ambas se califican en una escala del 1 al 5, permitiendo una medición objetiva y contextualizada que apoya la priorización y gestión efectiva de los riesgos.

1. **Probabilidad de ocurrencia**

La probabilidad refleja la frecuencia o posibilidad de que un riesgo específico se materialice durante la ejecución o explotación del proyecto. La asignación de un valor numérico se realiza considerando los siguientes factores**:**

* **Historial y estadísticas previas:** La experiencia acumulada en proyectos similares indica la recurrencia de ciertos riesgos. Por ejemplo, ataques comunes en aplicaciones web, como inyección SQL (SQL Injection) o Cross-Site Scripting (XSS), presentan alta probabilidad (valores 4 o 5) debido a su prevalencia y naturaleza frecuente.
* **Condiciones particulares del proyecto:** La implementación de protocolos de seguridad, como cifrado, autenticación multifactorial y controles de acceso robustos, reduce la probabilidad de riesgos asociados a fugas o accesos no autorizados, asignándoles valores bajos (1 o 2).
* **Factores externos:** La existencia de regulaciones, vulnerabilidades conocidas o amenazas emergentes puede elevar la probabilidad, al aumentar el riesgo de cumplimiento o exposición.
* **Madurez del equipo:** Equipos con experiencia y buenas prácticas disminuyen la ocurrencia de errores humanos, reduciendo la probabilidad.

1. **Impacto**

El impacto mide la severidad o daño potencial que el riesgo ocasionaría si llegara a ocurrir. Se valora atendiendo a:

* **Seguridad y confidencialidad:** Riesgos que comprometen datos sensibles, como acceso no autorizado a información de usuarios, reciben un alto impacto (4 o 5), dada la gravedad y consecuencias legales y éticas.
* **Operación:** Riesgos que puedan provocar interrupciones prolongadas o pérdida significativa de servicio se califican con impacto alto.
* **Costos económicos y legales:** Posibles multas, sanciones o demandas incrementan el impacto del riesgo.
* **Reputación y confianza:** Daños a la imagen institucional o pérdida de confianza de usuarios se consideran de alto impacto.
* **Recuperación:** Riesgos difíciles de mitigar o revertir reciben una valoración elevada por su complejidad y costos asociados.

**Sección 2 .**

**Web Content Accessibility Guidelines seccion 2.1 WCAG**

Es un estándar internacional desarrollado por el **W3C (World Wide Web Consortium)** que establece pautas para que los contenidos web sean **accesibles para todas las personas**, incluidas aquellas con **discapacidad visual, auditiva, motora, cognitiva, del habla o neurológica**.

**¿Qué dice WCAG 2?1?**

WCAG 2.1 se basa en **cuatro principios fundamentales**:

1. **Perceptible**: La información debe ser presentada de forma que los usuarios puedan percibirla (por ejemplo, usando texto alternativo en imágenes, contraste suficiente entre texto y fondo, subtítulos en multimedia).
2. **Operable**: La interfaz debe ser funcional con teclado, no provocar convulsiones, y facilitar la navegación (como saltos a contenido, botones accesibles, etc.).
3. **Comprensible**: El contenido y la interfaz deben ser legibles, comprensibles y predecibles.
4. **Robusto**: El contenido debe ser lo suficientemente robusto para ser interpretado correctamente por diferentes agentes de usuario, incluidos lectores de pantalla y navegadores. (“Accesibilidad: Definición y su Importancia en el Entorno Digital”)

Además, WCAG 2.1 **amplía WCAG 2.0** con nuevas recomendaciones para **dispositivos móviles, accesibilidad táctil, visión baja y discapacidades cognitivas**.

**Justificación de uso en los requerimientos funcionales**

**Motivo de inclusión en el proyecto**:

La inclusión de la **norma WCAG 2.1** dentro de los **requerimientos funcionales** del proyecto tiene como objetivo **garantizar que el sistema web sea accesible e incluyente**, en cumplimiento con las buenas prácticas internacionales de accesibilidad y con posibles exigencias legales nacionales (como la Ley Federal para la Inclusión de las Personas con Discapacidad en México).

En el desarrollo del sistema se consideraron los lineamientos establecidos por la norma **WCAG 2.1** (Web Content Accessibility Guidelines), emitida por el **W3C**, con el objetivo de asegurar que la interfaz de usuario sea accesible para personas con distintos tipos de discapacidad visual, motriz o cognitiva.

Estas directrices se integraron dentro de los **requerimientos funcionales** para cumplir con los principios de **perceptibilidad, operabilidad, comprensibilidad y robustez**, priorizando aspectos como el uso de contraste adecuado, navegación por teclado, etiquetas alternativas, estructuras semánticas claras y compatibilidad con tecnologías de asistencia.

Esta decisión se fundamenta en la necesidad de promover un diseño **universal e incluyente**, optimizar la experiencia del usuario en distintos dispositivos y contextos, y alinear el sistema con estándares internacionales que fortalecen la **calidad, legalidad y sostenibilidad del desarrollo web**.

**Sección 3 .**

**Protección contra ataques comunes según OWASP Top 10**

Como parte de las estrategias para garantizar la **seguridad integral del sistema**, se adoptaron las **recomendaciones establecidas en el estándar OWASP Top 10 (2021)**, que representa el consenso global sobre las vulnerabilidades más críticas que afectan a aplicaciones web.

La integración de estas prácticas en el desarrollo se justificó por la necesidad de **proteger los datos sensibles de los usuarios, prevenir accesos no autorizados, mitigar riesgos de corrupción de información y garantizar el cumplimiento con estándares de seguridad reconocidos a nivel internacional**.

Cada módulo del sistema fue diseñado e implementado tomando en cuenta controles y validaciones que abordan los siguientes vectores de riesgo identificados por OWASP:

* **A01:2021 - Control de acceso roto:** se implementaron validaciones a nivel de backend, control de roles y autenticación segura.
* **A02:2021 - Fallas criptográficas:** se aplicó cifrado en contraseñas y tokens de sesión (bcrypt, JWT).
* **A03:2021 - Inyecciones (SQL/XSS):** se sanitizaron entradas y se usaron consultas preparadas en bases de datos.
* **A05:2021 - Configuración de seguridad incorrecta:** se aplicaron políticas CORS, headers HTTP de seguridad y eliminación de configuraciones por defecto.
* **A07:2021 - Falla de identificación/autenticación:** uso de tokens seguros, tiempo de expiración y bloqueo tras intentos fallidos.
* **A09:2021 - Fallas en el registro y monitoreo:** se habilitó logging de eventos sospechosos y alertas en tiempo real.

El cumplimiento de estas prácticas reduce significativamente la superficie de ataque del sistema, incrementa la **resiliencia ante ciberataques** y asegura la **conformidad con normativas actuales de ciberseguridad**, como la Ley Federal de Protección de Datos Personales en Posesión de los Particulares (México).

**Relación entre OWASP Top 10 y medidas de mitigación implementadas**

|  |  |  |
| --- | --- | --- |
| Nº | Vulnerabilidad OWASP Top 10 (2021) | Medida(s) implementada(s) en el sistema |
| A01 | Broken Access Control | - Control estricto de roles y permisos- Middleware de autenticación- Validación de accesos en backend |
| A02 | Cryptographic Failures | - Uso de TLS 1.2+ para datos en tránsito- Hashing con bcrypt para contraseñas- AES-256 para datos sensibles en reposo |
| A03 | Injection (SQL, NoSQL, OS, etc.) | - Consultas parametrizadas (ORM / SQLAlchemy / prepared statements)- Escapado de entradas- Validación estricta de datos |
| A04 | Insecure Design | - Seguridad desde la arquitectura- Modelado de amenazas- Uso de principios como “menos privilegios” |
| A05 | Security Misconfiguration | - Headers de seguridad (CSP, X-Frame-Options)- Eliminación de servicios no utilizados- Actualización continua |
| A06 | Vulnerable and Outdated Components | - Gestión activa de dependencias- Escaneo de seguridad automatizado (dependabot, Snyk)- CI/CD con validaciones |
| A07 | Identification and Authentication Failures | - Uso de JWT seguros y tokens de renovación- Autenticación multifactor- Política de bloqueo ante intentos fallidos |
| A08 | Software and Data Integrity Failures | - Integridad de datos verificada con hashes- Firmas de integridad en paquetes de actualización |
| A09 | Security Logging and Monitoring Failures | - Integración con herramientas como Grafana, Kibana y Prometheus- Alertas ante eventos críticos |
| A10 | Server-Side Request Forgery (SSRF) | - Validación de URLs externas- Lista blanca de destinos confiables- Desactivación de redirecciones inseguras |

**Sección 4 .**

**Encriptación segura de datos sensibles en tránsito y en reposo**

Con el fin de asegurar la **confidencialidad, integridad y disponibilidad de los datos sensibles** manejados por el sistema, se definió como requisito no funcional obligatorio el uso de **encriptación robusta tanto en tránsito como en reposo**, siguiendo los principios establecidos por los estándares internacionales de seguridad de la información como **ISO/IEC 27001** y las **recomendaciones del Instituto Nacional de Estándares y Tecnología (NIST)**.

La encriptación en **tránsito** se implementó mediante el uso de **protocolo TLS (Transport Layer Security)** en su versión 1.2 o superior, garantizando que toda la comunicación entre cliente y servidor (por ejemplo, formularios, autenticación, consultas de datos) esté cifrada y protegida contra ataques como *man-in-the-middle (MITM)*, *sniffing* y *spoofing*. Esto se reforzó con el uso de certificados SSL válidos, renovación automatizada y políticas de red estrictas (HSTS).

En cuanto a la encriptación de datos **en reposo**, se utilizó:

* **Hashing fuerte (bcrypt)** para contraseñas de usuario.
* **Cifrado AES-256** para almacenar archivos o información sensible en bases de datos o discos.
* Cifrado de variables sensibles dentro del entorno de ejecución mediante secretos gestionados en contenedores seguros (por ejemplo, env cifrados y monitoreados).

Estas prácticas aseguran que incluso en caso de una brecha física o lógica, los datos robados no puedan ser leídos o reutilizados. Además, se cumplen con los principios de **seguridad por diseño**, **privacidad por defecto** y **responsabilidad proactiva**, en alineación con la **Ley Federal de Protección de Datos Personales en Posesión de los Particulares** (México).

La implementación de estos mecanismos garantiza una arquitectura tecnológica robusta, alineada con los estándares de mejores prácticas en ciberseguridad, y reduce sustancialmente el riesgo de sanciones legales o pérdida de confianza por parte de los usuarios finales.

**Aplicación de encriptación por tipo de dato sensible**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de Dato Sensible | Estado del Dato | Técnica de Encriptación / Protección Utilizada | Herramientas / Tecnologías Aplicadas | Justificación de Seguridad |
| Contraseñas | En reposo | Hashing con bcrypt (salt incluido) | bcrypt, passlib, Argon2 | Evita reversibilidad; protege incluso ante brechas de BD |
| Tokens de sesión (JWT) | En tránsito y reposo | Firmado y cifrado con clave secreta (HS256 / RS256) | JWT (jsonwebtoken, PyJWT) | Previene manipulación y falsificación de identidad |
| Datos personales del usuario | En reposo | Cifrado simétrico con AES-256 | PyCrypto, cryptography | Garantiza confidencialidad de nombre, correo, teléfono, etc. |
| Formularios de contacto/registro | En tránsito | TLS 1.2+ con certificado SSL | HTTPS, Let’s Encrypt | Previene sniffing/intercepción de datos |
| Archivos cargados (documentos) | En reposo | Cifrado por bloques (AES) con claves rotativas | cryptography. fernet, almacenamiento seguro | Protección ante accesos no autorizados o filtraciones |
| Base de datos completa | En reposo | Transparent Data Encryption (TDE) | PostgreSQL, MySQL TDE, Azure SQL, etc. | Protección a nivel físico del almacenamiento |
| Cookies de autenticación | En tránsito y reposo | Atributos HttpOnly, Secure, SameSite=Strict | Headers HTTP configurados desde backend/frontend | Protege tokens ante ataques XSS y evita envío no intencional a sitios externos |