

**GASIC 8– Seguridad en el Desarrollo y Adquisición de Software**

**Preguntas de Auditoría**

**Funciones y Responsabilidades**

**Competencias Básicas del Auditor**

1. **Conocimiento Profundo en Desarrollo de Software Seguro:** Entender los principios de seguridad en el desarrollo de software, incluyendo buenas prácticas de codificación, diseño seguro, y consideraciones de seguridad en el ciclo de vida del desarrollo.
2. **Experiencia en Identificación de Vulnerabilidades:** Tener habilidades para identificar y evaluar vulnerabilidades en el código y el diseño del software, incluyendo análisis estático y dinámico, pruebas de penetración y revisión de arquitecturas.
3. **Conocimiento en Criptografía y Protección de Datos:** Comprender los principios de la criptografía y cómo se aplica en la protección de datos sensibles dentro del software, incluyendo la gestión segura de claves y la integridad de los datos.
4. **Habilidades en Pruebas de Seguridad y Evaluación de Riesgos:** Ser capaz de realizar pruebas de seguridad exhaustivas para evaluar la resistencia del software frente a posibles ataques, así como evaluar los riesgos asociados con el software.
5. **Conocimientos en Cumplimiento y Normativas:** Estar familiarizado con los requisitos de cumplimiento y normativas relevantes en el ámbito de la seguridad del software, como GDPR, PCI DSS, y normas ISO/IEC.
6. **Capacidad Analítica:** Ser capaz de analizar datos de seguridad y tendencias para identificar posibles amenazas y vulnerabilidades emergentes en el software.
7. **Habilidades de Comunicación Técnica:** Ser capaz de comunicar de manera efectiva los hallazgos de seguridad a los desarrolladores, gerentes y otros interesados, y proponer recomendaciones claras y prácticas.
8. **Conocimientos en Gestión de Incidentes y Respuesta a Incidentes:** Entender los procedimientos y prácticas para la gestión de incidentes de seguridad en el software, incluyendo la capacidad de respuesta rápida y eficaz.

# Visión General

Definir roles específicos, integrar responsabilidades de seguridad y capacitar a los involucrados son prácticas esenciales para establecer un entorno colaborativo y eficiente. Desde el punto de vista de un auditor, estas acciones deben ser documentadas, revisadas periódicamente y alineadas con los objetivos de seguridad de la organización para garantizar su efectividad.

**1.1 Asignación y Comprensión de Roles**

Las funciones y responsabilidades deben asignarse de manera formal, asegurando que las personas designadas comprendan y acepten su autoridad y tareas específicas. Esto incluye integrar roles de seguridad dentro de los equipos de desarrollo de software, estableciendo un enlace directo entre las actividades de seguridad y el ciclo de vida del desarrollo. Un auditor evaluará si estas asignaciones están claramente documentadas y si las personas involucradas entienden cómo su papel impacta la seguridad general del proyecto.

**1.2 Capacitación y Formación**

Definir y planificar la capacitación requerida para cada rol es fundamental para asegurar que el personal cuente con las habilidades necesarias para cumplir con sus responsabilidades. La organización debe crear programas de formación adaptados a sus necesidades, documentando los resultados esperados para cada rol y midiendo el impacto de la capacitación para identificar áreas de mejora. Desde una perspectiva de auditoría, la evidencia de programas de formación efectivos y actualizados es un indicador clave del compromiso organizacional con la seguridad.

**1.3 Monitoreo y Revisión de Funciones**

Las funciones y responsabilidades asignadas deben revisarse anualmente para garantizar que sigan siendo relevantes y efectivas frente a los cambios organizacionales o tecnológicos. Además, la implementación de herramientas y procesos que promuevan la comunicación entre roles relacionados con el SDLC facilita la colaboración y minimiza errores en el desarrollo seguro. Un auditor verificará si estas revisiones se realizan de manera sistemática y si las herramientas implementadas mejoran la interacción entre los equipos.

**1.4 Liderazgo y Concienciación**

Es crucial designar un líder o equipo responsable del desarrollo seguro, que actúe como enlace entre la alta dirección y los equipos operativos. Este liderazgo debe fomentar la concienciación sobre los riesgos asociados con el desarrollo sin seguridad integrada y educar al personal sobre el compromiso organizacional con la seguridad. Desde el punto de vista de un auditor, evaluar el impacto de estas iniciativas permite determinar si la organización está creando una cultura sólida de seguridad en el desarrollo de software.

**1.5 Propiedad y Responsabilidad del Código**

Cada proyecto debe contar con un grupo designado como propietario del código, que asuma la responsabilidad de su gestión y seguridad. Esto incluye supervisar cambios, garantizar la calidad y asegurar que los estándares de seguridad se apliquen consistentemente. Un auditor revisará si esta asignación está formalizada y si existen métricas para evaluar el desempeño del propietario del código en términos de cumplimiento de los objetivos de seguridad.

# Modelo de Madurez

|  |  |
| --- | --- |
| NIVEL 1 [Impredecible y reactivo]: La actividad se completa, pero a menudo se retrasa y supera el presupuesto. | 1. Asignar la responsabilidad y la autoridad globalmente para realizar el proceso y las tareas específicas del proceso.  1.1 Confirmar que las personas a las que se ha asignado la responsabilidad y la autoridad las comprenden y las aceptan. 2. Integrar roles de seguridad en el equipo de desarrollo de software. |
| NIVEL 2 [Gestionado a nivel de proceso]: Las actividades se planifican, ejecutan, miden y controlan. | 1. Documentar los resultados deseados de la capacitación para cada rol. 2. Definir el tipo de capacitación requerida para cada rol. 3. Crear un plan de formación para cada rol. 4. Adquirir o crear capacitación adaptada a la organización. |
| NIVEL 3 [Proactivo antes que reactivo]: Los estándares de toda la organización brindan orientación a través de proyectos, programas y carteras. | 1. Realizar una revisión anual de todas las funciones y responsabilidades. 2. Implementar herramientas y procesos para promover la comunicación entre los roles relacionados con SDLC. 3. Designar un grupo de personas como propietario del código para cada proyecto. |
| NIVEL 4 [Gestionado Cuantitativamente]: La organización está basada en datos con objetivos cuantitativos de mejora del desempeño que son predecibles | 1. Medir el desempeño de los resultados de la capacitación para identificar áreas de mejora. |
| NIVEL 5 [Optimizado]: La organización se centra en la mejora continua y está diseñada para girar y responder a las oportunidades y los cambios. | 1. Designar un líder o equipo responsable del proceso de desarrollo seguro. 2. Aumentar la conciencia sobre los riesgos de desarrollo de software sin seguridad integrada. 3. Ayudar a la alta dirección a comunicar el apoyo al desarrollo seguro. 4. Educar a todo el personal sobre el compromiso de la alta dirección con el desarrollo seguro. |

1. **Preguntas de Auditoría para Funciones y Responsabilidades**

|  |
| --- |
| Temática: Definición de Funciones y Responsabilidades |
| La correcta implementación de funciones y responsabilidades en el desarrollo y adquisición de software es un pilar fundamental para garantizar la seguridad y eficacia de los sistemas .Esta dimensión crítica del proceso no solo establece roles específicos para los participantes, sino que también define las responsabilidades asociadas con la seguridad de los activos digitales y la mitigación de riesgos. La importancia de auditar este aspecto reside en su capacidad para validar el cumplimiento normativo, identificar brechas de seguridad, asegurar responsabilidades claras, evaluar competencias, optimizar recursos y fomentar una cultura organizacional que tenga en cuenta la seguridad de sus activos. A través de esta auditoría, se busca no solo cumplir con estándares y regulaciones, sino también fortalecer la resiliencia del equipo frente a amenazas en constante evolución y fomentar la mejora continua en todas las fases del ciclo de vida del software.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Existe un proceso formal para asignar responsabilidades y autoridades en el desarrollo y adquisición de software?   + ¿Se confirma periódicamente que las personas asignadas comprenden y aceptan sus responsabilidades y autoridades?   + ¿Las responsabilidades y autoridades asignadas abarcan todo el proceso y tareas específicas relevantes?   + ¿Se han integrado roles específicos de seguridad en el equipo de desarrollo de software?   + ¿Existe claridad sobre las funciones y responsabilidades de los roles de seguridad dentro del equipo?     - ¿Estos roles tienen la autoridad necesaria para influir en decisiones de seguridad a lo largo del ciclo de vida del software?   + ¿Se promueve la colaboración efectiva entre los roles de seguridad y otros miembros del equipo? * ¿Están documentados claramente los resultados deseados de la capacitación para cada rol en el desarrollo y adquisición de software?   + ¿Esta documentación se comparte y comunica de manera efectiva a las partes interesadas?   + ¿Se ha definido claramente el tipo de capacitación requerida para cada rol involucrado en el proceso?   + ¿Se consideran aspectos específicos de seguridad en el diseño del plan de capacitación?   + ¿Cada rol tiene un plan de formación adaptado a sus responsabilidades y requisitos específicos?   + ¿Este plan de formación se actualiza de manera regular para adaptarse a cambios en el entorno de seguridad o en el equipo? * ¿Existe un proceso para evaluar y mejorar continuamente los recursos de capacitación disponibles?   + ¿Esta revisión abarca todos los roles clave y se actualiza según sea necesario? * ¿Se han implementado herramientas y procesos efectivos para fomentar la comunicación entre los roles relacionados con el ciclo de vida del desarrollo de software?   + ¿Existen canales claros de comunicación para abordar cuestiones de seguridad durante todo el SDLC? * ¿Cada proyecto tiene designado un grupo de personas como propietario del código?   + ¿Se definen claramente las responsabilidades del propietario del código en términos de seguridad?   + ¿Se mide el desempeño de los resultados de la capacitación de manera regular?     - ¿Las áreas de mejora identificadas se abordan de manera proactiva en la planificación de la capacitación futura? * ¿Se ha designado un líder o equipo específico responsable del proceso de desarrollo seguro?   + ¿Este líder o equipo tiene la autoridad necesaria para influir en decisiones relacionadas con la seguridad durante todo el SDLC?   + ¿Se llevan a cabo iniciativas para aumentar la conciencia sobre los riesgos asociados con el desarrollo de software sin seguridad integrada?     - ¿Se han implementado medidas para mitigar estos riesgos y fomentar prácticas de desarrollo seguro? * ¿La alta dirección ha designado un líder o equipo para respaldar y liderar el desarrollo seguro?   + ¿Se comunica de manera efectiva el compromiso de la alta dirección con el desarrollo seguro a todo el personal?   + ¿Se lleva a cabo una campaña educativa para informar a todo el personal sobre el compromiso de la alta dirección con el desarrollo seguro?   + ¿Esta educación abarca la importancia de la seguridad en el desarrollo de software y cómo cada miembro del equipo contribuye a este objetivo? |