

**09 – Acceso a Código Fuente**

**CASIG 4 – Control de Acceso e Identidades**

**Preguntas de Auditoría**

**Acceso a Código Fuente**

|  |
| --- |
| **Competencias Básicas del Auditor**   1. **Conocimiento de Normativas y Estándares**: Familiaridad con marcos de referencia y estándares de la industria como NIST, ISO 27001, CIS, entre otros. 2. **Habilidades Analíticas**: Capacidad para analizar registros, logs y datos para identificar patrones anómalos o actividades sospechosas. 3. **Comprensión de Sistemas de Autenticación**: Entender diferentes mecanismos de autenticación, como autenticación de dos factores, tokens, certificados digitales, entre otros. 4. **Conocimiento de Tecnologías de Red**: Entender cómo funcionan las redes, protocolos y cómo se pueden explotar las vulnerabilidades relacionadas con el acceso. 5. **Habilidades de Comunicación**: Capacidad para comunicar hallazgos, riesgos y recomendaciones de manera clara a diferentes audiencias, desde técnicos hasta ejecutivos. 6. **Conocimiento de Herramientas de Auditoría**: Familiaridad con herramientas y software específicos utilizados para auditar sistemas y redes. 7. **Comprensión de la Gestión de Riesgos**: Capacidad para identificar, evaluar y priorizar riesgos asociados con la administración de cuentas y credenciales. 8. **Habilidades de Investigación**: Capacidad para investigar incidentes de seguridad, identificar la causa raíz y recomendar medidas correctivas. 9. **Conocimiento de Procesos de Negocio**: Entender cómo funcionan los procesos de negocio y cómo la administración de cuentas y credenciales se integra en ellos. 10. **Habilidades Interpersonales**: Capacidad para trabajar en equipo, gestionar conflictos y colaborar con diferentes departamentos y partes interesadas. 11. **Conocimiento de Legislación y Cumplimiento**: Estar al tanto de las leyes y regulaciones pertinentes que afectan la administración de cuentas y credenciales en la jurisdicción relevante. |

# Visión General

El acceso al código fuente, así como a las herramientas de desarrollo y bibliotecas de software, se gestiona de forma rigurosa para proteger la integridad y seguridad de los sistemas de información. Este control incluye la administración de permisos de lectura y escritura según el rol y las necesidades empresariales, restringiendo el acceso para minimizar riesgos de modificación o acceso no autorizado. Las organizaciones centralizan el almacenamiento del código fuente en sistemas de gestión especializados y aplican controles de acceso específicos a bibliotecas y herramientas de desarrollo. En entornos compartidos o que usan componentes de código abierto, el acceso de lectura puede permitirse de forma más amplia, mientras que el acceso de escritura sigue siendo altamente controlado.

**1.1 Directrices para el Acceso Controlado a Código Fuente**

Para reducir el riesgo de alteración o corrupción, el acceso se regula mediante procedimientos establecidos que administran el acceso directo a través de herramientas de desarrollo que registran las actividades y permisos. Además, la actualización y autorización de cambios en el código fuente se gestionan mediante procesos formales de control de cambios. La documentación del acceso y los cambios permite una trazabilidad continua y una auditoría exhaustiva de todas las interacciones con el código fuente, proporcionando una capa adicional de seguridad y transparencia.

**1.2 Perspectiva desde la Auditoría en Control de Código Fuente**

Desde el ámbito de la auditoría en SIC, se evalúa la efectividad de los controles sobre el acceso y modificación del código fuente, revisando los registros de auditoría y los procesos de autorización de cambios. Estos controles aseguran que solo personal autorizado accede o modifica el código, lo cual es clave para mantener la integridad y protección de la infraestructura de desarrollo y sus aplicaciones empresariales.

En Chile los softwares están protegidos por la ley 17.336 que regula la Propiedad Intelectual y los derechos de autor de todos aquellos creadores de obras artísticas, literarias y científicas.

|  |
| --- |
| * Ley 17.336-Propiedad Intelectual:   Artículo 1°- La presente ley protege los derechos que, por el solo hecho de la creación de la obra, adquieren los autores de obras de la inteligencia en los dominios literarios, artísticos y científicos, cualquiera que sea su forma de expresión, y los derechos conexos que ella determina. El derecho de autor comprende los derechos patrimonial y moral, que protegen el aprovechamiento, la paternidad y la integridad de la obra. |

# Modelo de Madurez

|  |  |
| --- | --- |
| NIVEL 1 [Impredecible y reactivo]: La actividad se completa, pero a menudo se retrasa y supera el presupuesto. | 1. El acceso de lectura y escritura al código fuente, las herramientas de desarrollo y las bibliotecas de software debe administrarse adecuadamente.  2. El acceso al código fuente y elementos asociados (como diseños, especificaciones, planes de verificación y planes de validación) y herramientas de desarrollo (por ejemplo, compiladores, constructores, herramientas de integración, plataformas y entornos de prueba) debe controlarse estrictamente.  3. Para el código fuente, esto se puede lograr controlando el almacenamiento central de dicho código, preferiblemente en el sistema de gestión de código fuente.  4. El acceso de lectura y el acceso de escritura al código fuente pueden variar según la función del personal.   5. Cuando los componentes del código son utilizados por varios desarrolladores dentro de una organización, se debe implementar el acceso de lectura a un repositorio de código centralizado. Además, si dentro de una organización se utilizan código fuente abierto o componentes de código de terceros, se puede proporcionar acceso de lectura a dichos repositorios de código externos en general. Sin embargo, el acceso de escritura aún debe estar restringido.  6. Si se pretende publicar el código fuente del programa, se deben considerar controles adicionales para garantizar su integridad (p. ej., firma digital).  7. Si el acceso al código fuente no se controla adecuadamente, el código fuente puede modificarse o algunos datos en el entorno de desarrollo (por ejemplo, copias de datos de producción, detalles de configuración) pueden ser recuperados por personas no autorizadas. |
| NIVEL 2 [Gestionado a nivel de proceso]: Las actividades se planifican, ejecutan, miden y controlan. | 1. Considerar las siguientes pautas para controlar el acceso a las bibliotecas de fuentes de programas con el fin de reducir el potencial de corrupción de los programas de computadora:  1.1 Administrar el acceso al código fuente del programa y las bibliotecas fuente del programa de acuerdo con los procedimientos establecidos;  1.2 Otorgar acceso de lectura y escritura al código fuente en función de las necesidades de la empresa y administrado para abordar los riesgos de alteración o uso indebido y de acuerdo con los procedimientos establecidos;  1.3 Actualización del código fuente y elementos asociados y otorgamiento de acceso al código fuente de acuerdo con los procedimientos de control de cambios y solo realizarlo después de haber recibido la autorización correspondiente;  1.4 No otorgar a los desarrolladores acceso directo al repositorio del código fuente, sino a través de herramientas para desarrolladores que controlan las actividades y autorizaciones en el código fuente;  1.5 Mantener las listas de programas en un entorno seguro, donde el acceso de lectura y escritura debe administrarse y asignarse adecuadamente;  1.6 Mantener un registro de auditoría de todos los accesos y de todos los cambios en el código fuente. |
| NIVEL 3 [Proactivo antes que reactivo]: Los estándares de toda la organización brindan orientación a través de proyectos, programas y carteras. |  |
| NIVEL 4 [Gestionado Cuantitativamente]: La organización está basada en datos con objetivos cuantitativos de mejora del desempeño que son predecibles |  |
| NIVEL 5 [Optimizado]: La organización se centra en la mejora continua y está diseñada para girar y responder a las oportunidades y los cambios. |  |

# Preguntas de Auditoría para el Acceso a Código de Fuente

|  |
| --- |
| 3.1 Temática: Políticas específicas, procedimiento y capacitación del personal para el control del Acceso al Código Fuente |
| Revisar y evaluar las medidas establecidas por una organización para garantizar la seguridad y la gestión adecuada de su código fuente, este punto tiene como objetivo asegurarse de que se estén aplicando prácticas adecuadas para controlar quiénes pueden acceder, modificar y gestionar el código fuente de un software o sistema. Las responsabilidades deben estar adecuadamente separadas para evitar conflictos de interés y reducir el riesgo de abuso de acceso al código fuente. Es necesario revisar la existencia de procedimientos de respaldo y recuperación en caso de pérdida o daño del código fuente.  Es importante revisar las políticas específicas establecidas que definen quiénes tiene los permisos para acceder al código fuente, esto incluye roles y niveles de autorización, así como restricciones basadas en la necesidad de conocer.  Verificar si los desarrolladores y miembros del equipo siguen los procedimientos y políticas establecidos al interactuar con el código fuente, incluye la revisión de registros de acceso y cambios para asegurarse de que se estén siguiendo las pautas.  Por último, se evalúa si la organización ofrece programas de capacitación periódicos sobre las mejores prácticas de seguridad al trabajar con el código fuente de softwares. Se verifica si el contenido de la capacitación es relevante y si se evalúa la comprensión de los empleados después de la formación.   * ¿La organización estable y comunica una política clara y específica sobre el acceso y las credenciales al código fuente? * Revise la documentación de la política para asegurarse de que esté formalmente establecida y que contenga detalles sobre quiénes pueden acceder, cómo se otorgan las credenciales y cuáles son los procedimientos para solicitar acceso. * ¿La política establece los roles y responsabilidades relacionados con el acceso al código fuente? * ¿Se describen claramente los roles y responsabilidades de las personas involucradas en el acceso al código fuente, como desarrolladores, administradores y revisores en la política? * ¿Existen mecanismos de difusión de la Política como Intranet, circulares informativas, correos electrónicos masivos o cualquier otro medio que la organización estime pertinente? * ¿Existen controles y políticas que regulan el acceso al código fuente abierto y a los componentes de terceros? * ¿Existe un equipo designado y claramente definido responsable de evaluar y otorgar accesos al código fuente? * ¿El equipo responsable de otorgar accesos al código fuente cuenta con los conocimientos y habilidades adecuados para evaluar las solicitudes de acceso? * Evalúe los registros de capacitación y certificaciones obtenidas por los miembros del equipo relacionadas con la seguridad de la información y el control de acceso. * Revise los criterios y requisitos establecidos en la documentación para evaluar las solicitudes de acceso, como la necesidad de conocer, el nivel de autorización requerido, etc. * ¿Están los niveles de acceso al código fuente definidos en función de roles específicos o responsabilidades dentro de la organización? * ¿Existen políticas o directrices claras que establezcan las diferencias en los niveles de acceso al código fuente según las funciones del personal? * ¿Se utilizan controles de autenticación y autorización para asegurar que los empleados solo tengan acceso al código fuente que es relevante para sus funciones? * ¿Existe un proceso formal para solicitar y otorgar acceso de lectura y escritura al código fuente, así como a las herramientas de desarrollo y bibliotecas de software? * ¿Se remueve los derechos de acceso para el personal que ya no trabaje en la institución? * ¿La organización tiene políticas o procedimientos establecidos para garantizar la integridad del código fuente en caso de que se pretenda publicarlo? * Evalúe una muestra de código fuente previamente publicado para determinar si se ha aplicado algún control adicional de integridad. * ¿Se utiliza la firma digital u otros métodos de autenticación para asegurar la integridad del código fuente antes de su publicación? * ¿Se administran los accesos al código fuente y las bibliotecas fuente de acuerdo con los procedimientos establecidos? * Verifique si se otorga acceso de lectura y escritura al código fuente basado en las necesidades de la Organización y se gestionan los riesgos adecuadamente. * Evalúe cómo los desarrolladores acceden al código fuente para verificar si se utilizan las herramientas de control apropiadas. * ¿Se mantienen listas de programas en un entorno seguro y se administran adecuadamente los accesos de lectura y escritura? * ¿Existe un programa de capacitación establecido para los desarrolladores y miembros del equipo sobre las mejores prácticas de seguridad al trabajar con el código fuente? * ¿El programa de capacitación se lleva a cabo periódicamente y de manera regular? * ¿El contenido del programa de capacitación aborda adecuadamente las mejores prácticas de seguridad al trabajar con el código fuente? * Obtenga una copia del material de capacitación utilizado y verificar si aborda conceptos y técnicas de seguridad en el contexto del desarrollo y manejo de código fuente. * ¿Se evalúa la comprensión y aplicación de las mejores prácticas de seguridad por parte de los participantes después de la capacitación? * ¿Existen procedimientos documentados para realizar respaldos periódicos del código fuente? * ¿Se establecen ubicaciones seguras y adecuadas para almacenar los respaldos del código fuente? * Realice una inspección física de las instalaciones para verificar la existencia de sistemas de almacenamiento seguro, como servidores de respaldo o medios de almacenamiento externos. * ¿Se realizan pruebas periódicas de restauración de los respaldos del código fuente? |

|  |
| --- |
| **3.2 Temática: Monitoreo y datos cuantitativos para evaluar el Acceso al Código Fuente** |
| Esta prueba implica examinar la efectividad de los sistemas y procesos implementados para supervisar quién accede al código fuente, cuándo lo hace y qué actividades realiza. Esto se logra a través de una evaluación basada en pruebas específicas diseñadas para medir la seguridad y la adecuación de los controles de acceso al código fuente. Se debe evaluar si la organización implementa el seguimiento de sesiones, lo que permite rastrear todas las acciones realizadas durante una sesión de acceso al código fuente.  Verificar si se implementan alertas automáticas para actividades inusuales o potencialmente sospechosas, como múltiples intentos de acceso fallidos o cambios masivos en el código. Examinar si se realizan análisis de tendencias en los datos cuantitativos para identificar patrones de acceso o actividad inusual a lo largo del tiempo y si se generan reportes de auditoría regulares basados en estos datos y los registros de acceso para detectar anomalías o tendencias preocupantes.   * ¿Se mantienen registros detallados de los usuarios que tienen acceso de lectura y escritura al código fuente y las herramientas de desarrollo? * Revise los registros de acceso y registros de auditoría que indiquen quién ha accedido al código fuente, elementos asociados y herramientas de desarrollo, así como cuándo y por qué motivo. * Los controles de acceso deberán contar con revisiones periódicas para validar su vigencia. * ¿La organización tiene establecido un proceso de monitoreo para detectar posibles riesgos en el código fuente? * ¿Se utilizan herramientas y técnicas de seguridad de la información para identificar posibles vulnerabilidades en el código fuente? * Para la evaluación, el auditor debe tener en consideración las siguientes Metodologías: * Microsoft SDL (Microsoft Security Development Lifecycle) * CLASP (Comprehensive, Lightweight Application Security Process) * OpenSAMM (Open Software Assurance Maturity Model) * Touchpoints * Para la evaluación, el auditor debe tener en consideración las siguientes Herramientas: * GitHub * SonarQube * Review Board * OWASP ZAP * DeepScan * Code Climate * Wapiti * Para la evaluación, el auditor debe tener en consideración las siguientes técnicas y mecanismos para el análisis estático: * Grep análisis * Propagación constante * Data-flow analysis * Búsqueda de patrones * ¿La organización mantiene un registro de auditoría de todos los accesos al código fuente? * Revise los accesos con una muestra aleatoria de registros de auditoría para comprobar si se registran los detalles de acceso, como la identificación del usuario, fecha, hora y propósito de acceso. * ¿El registro de auditoría de accesos al código fuente incluye toda la información completa y detallada? * ¿La organización mantiene un registro de auditoría de todos los cambios realizados en el código fuente? * ¿La organización tiene un sistema de gestión de código fuente (SCM) designado para centralizar y controlar el almacenamiento del código fuente? * Verifique la existencia de repositorios que almacenan el código fuente a través del SCM. * ¿Existe un procedimiento establecido para el control de cambios en el código fuente? * Todos los cambios realizados, ya sean, modificaciones, mejoras o actualizaciones que se efectúen al código, deben estar debidamente documentados y registrados en una bitácora y en los archivos LOG. * Todos los cambios, modificaciones, mejoras o actualizaciones que se efectúen al código, se realizan con la debida autorización del personal responsable. * ¿Se ha implementado un repositorio de código centralizado para almacenar los componentes compartidos utilizados por varios desarrolladores? * ¿El acceso de lectura al repositorio de código centralizado está limitado a los desarrolladores autorizados? * Revise la lista de usuarios o roles que tienen acceso de lectura al repositorio centralizado y compararla con una lista de desarrolladores autorizados. * ¿Se recopilan datos cuantitativos de manera regular para evaluar el proceso de administración del acceso al código fuente? * Verifique de datos cuantitativos que se recopilan para evaluar la administración del acceso al código fuente, como: * Tiempo promedio de aprobación de acceso al código fuente * Tasa de solicitudes rechazadas * Frecuencia de cambios no autorizados en el código fuente * Tiempo promedio para revocar accesos a usuarios desvinculados * Tasa de cumplimiento de auditoría en el acceso al código fuente * Evalúe si existen mecanismos o procesos documentados para analizar los   datos cuantitativos con el fin de identificar áreas que requieran mejoras.   * ¿Las decisiones tomadas con respecto a la administración del acceso al código fuente están respaldadas por evidencia basada en los datos recopilados? * ¿El proceso de revisión y mejora está basado en los resultados de la monitorización y auditorías? * ¿Se lleva a cabo la monitorización regular de los privilegios de acceso al código fuente? * ¿Se realizan auditorías periódicas para evaluar la eficacia y cumplimiento de la Administración de los privilegios de acceso? * ¿Las acciones de mejora derivadas de las revisiones y auditorías se implementan y se realiza un seguimiento de su efectividad? |