ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“DISEÑO DE UN MARCO DE SEGURIDAD MEDIANTE EL USO DE ESTÁNDARES INTERNACIONALES PARA OPTIMIZAR PROCESOS DE DESARROLLO Y DESPLIEGUE DE SISTEMAS INFORMÁTICOS DENTRO DE UNA UNIVERSIDAD PÚBLICA”

TRABAJO DE TITULACIÓN

Previa a la obtención del Título de:

MAGÍSTER EN SEGURIDAD INFORMÁTICA

Presentado por:

ING. CARLOS ANDRÉS ORTEGA VENTURA

ING. CARLOS ANDRÉS GARZÓN TORRES

Guayaquil – Ecuador

2024

AGRADECIMIENTO

Agradezco sinceramente a todas las personas que han sido fundamentales en la realización de esta tesis:

A Dios por brindarme salud y sabiduría.

A mi esposa por su amor y paciencia durante todo este proceso.

A mis padres por todo su apoyo y ser mi ejemplo de vida.

A mis amigos que formaron parte de esta maestría.

A las autoridades de mi empresa por darme la oportunidad de tomar este desafío.

Ing. Carlos Andrés Ortega Ventura

AGRADECIMIENTO

Agradezco a Dios por darme fortaleza y sabiduría en este reto.

A nuestro Tutor por su orientación y apoyo en el desarrollo de nuestra tesis.

Las autoridades del departamento por ofrecerme la oportunidad de participar en esta maestría.

Ing. Carlos Andrés Garzón Torres

DEDICATORIA

Quiero dedicar este trabajo a las personas más importantes de mi vida:

A mi esposa, por estar siempre a mi lado y motivarme a crecer en mi carrera profesional.

A mis padres, por brindarme todo su apoyo para cumplir esta meta.

A mi hermano Samuel, por su afecto y carisma.

Ing. Carlos Andrés Ortega Ventura

DEDICATORIA

Quiero dedicar este trabajo a mi Madre Martha Germania Torres Cruz y sus hermanos, a mi Hermano Geovanny Gabriel Garzón Torres, quienes con su apoyo y sabiduría me incitaron a aceptar esta oportunidad para mejorar profesionalmente.

Ing. Carlos Andrés Garzón Torres

TRIBUNAL DE SUSTENTACIÓN

Mgtr. Lenin Freire Cobo

TUTOR DEL TRABAJO DE TITULACIÓN

Mgs. Juan Carlos García

REVISOR

DECLARACIÓN EXPRESA

La responsabilidad del contenido de esta Tesis de Postgrado nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.

Ing. Carlos Andrés Ortega Ventura

Ing. Carlos Andrés Garzón Torres

RESUMEN

El presente trabajo de titulación se centra en la elaboración de un marco de seguridad para la Dirección de Tecnologías de la Información (DTI) de una Institución de Educación Superior (IES) pública en Ecuador, con el objetivo de optimizar los procesos de desarrollo y despliegue de sus aplicaciones informáticas.

A lo largo de esta tesis, se ha identificado que los procesos de desarrollo y despliegue gestionados por la DTI presentan diversas deficiencias en materia de seguridad, tales como la ausencia de políticas formales de seguridad de la información, la falta de pruebas de seguridad en el ciclo de vida del desarrollo de software y la centralización del despliegue de aplicaciones hacia una sola persona. Estos problemas han expuesto a la institución a riesgos significativos, como la interrupción de servicios críticos.

Ante la creciente amenaza de ciberataques y para abordar estos desafíos, se propone la elaboración de políticas de seguridad basadas en la norma ISO/IEC 27002:2022 y el Marco de Desarrollo de Software Seguro (SSDF) de NIST. Estas normas incorporan controles tecnológicos y organizacionales alineados con las mejores prácticas internacionales. Entre los controles recomendados se encuentran la implementación de autenticación segura, codificación segura, separación de ambientes de desarrollo y producción, entre otros. Además, se

define una estructura de roles y responsabilidades para garantizar la correcta aplicación de estas políticas.

El marco propuesto no solo busca mejorar la seguridad de las aplicaciones informáticas que mantiene la DTI, sino también promover una cultura de seguridad dentro de la organización, asegurando que el personal esté adecuadamente capacitado y consciente de las prácticas de desarrollo seguro.

Con la adopción de estas políticas, se espera reducir significativamente los riesgos asociados al desarrollo de software y fortalecer la protección de los activos de información de la DTI y por ende de la IES.

ÍNDICE GENERAL

ABREVIATURAS Y SIMBOLOGÍAS

IES	Institución de Educación Superior
TI	Tecnología de Información
DTI	Dirección de Tecnologías de la Información
DDA	Departamento de Desarrollo de Aplicaciones
DI	Departamento de Infraestructura
DSR	Departamento de Seguridad y Redes
DST	Departamento de Soporte Técnico
ISO	Organización Internacional de Normalización
NIST	Instituto Nacional de Estándares y Tecnología
SSDF	Marco de Desarrollo de Software Seguro
MAGERIT	Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información
SGSI	Sistema de Gestión de la Seguridad de Información
SDLC	Ciclo de Vida del Desarrollo de Software
IDE	Entorno de Desarrollo Integrado

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad, las Instituciones de Educación Superior (IES) enfrentan desafíos relacionados con la seguridad de la información debido al creciente uso de tecnologías de la información (TI) y la dependencia de sistemas informáticos para la gestión de datos y operaciones académicas.

En este contexto, la protección de los activos de información se ha convertido en una prioridad ante el aumento de ciberataques que amenazan la integridad, disponibilidad y confidencialidad de la información. Esta situación expone a las IES a posibles ataques informáticos, tales como filtraciones de datos sensibles y ataques de denegación de servicio, que podrían afectar gravemente tanto a la reputación de la institución como a su operación financiera.

Por tal motivo, el propósito de este trabajo de titulación es diseñar un marco de seguridad para optimizar los procesos de desarrollo y despliegue de las aplicaciones informáticas gestionadas por la Dirección de Tecnologías de la Información (DTI) de una IES pública en Ecuador.

En este marco de seguridad, que comprende un conjunto de políticas, no solo se espera que cumpla con estándares internacionales, sino que también se adapte a las necesidades específicas de la DTI. Además, se busca garantizar la protección de los activos de información, fomentar una cultura de seguridad dentro de la organización y promover prácticas de desarrollo seguro.

En los siguientes capítulos, se detallarán los antecedentes y el contexto de la organización, se profundizará en los problemas que enfrenta la DTI, se analizarán los riesgos asociados al desarrollo y despliegue de aplicaciones informáticas, y se propondrán controles específicos para mejorar la seguridad de la información.

CAPÍTULO I

GENERALIDADES

1.1. Antecedentes

La organización sujeta al presente trabajo de titulación forma parte de una Institución de Educación Superior (IES) pública en Ecuador, la cual lleva a cabo procesos de desarrollo de software que incluyen operaciones de despliegue de sus aplicaciones. Estos procesos en cuestión son esenciales para el funcionamiento eficiente de la Dirección de Tecnologías de la Información (DTI), no obstante, la realidad es que la mayoría de estos procedimientos no se fundamentan en normativas internacionales reconocidas para la gestión de la seguridad de la información. Este hecho resalta la necesidad de adoptar un marco de referencia estandarizado que mejore la estructura y coherencia de las prácticas de seguridad, garantizando así una protección más robusta de los activos de información.

Actualmente se han multiplicado los ataques de ciberseguridad a instituciones gubernamentales y las IES no son la excepción. Una conclusión importante luego de la revisión de literatura consultada es que cuanto más sea la dependencia hacia el uso de internet en infraestructuras y tecnologías informáticas, mayor es la probabilidad de que los sistemas de información sean vulnerados mediante técnicas de intrusión, de manera que estas posibles vulnerabilidades representarían un riesgo a la seguridad parcial o total de los datos [1].

1.2. Descripción del Problema

El Departamento de Desarrollo de Aplicaciones (DDA) que forma parte de la DTI de una IES está enfrentando dificultades durante la fase de desarrollo y despliegue de sus aplicaciones o sistemas informáticos.

Es importante mencionar que la DTI debe gestionar principalmente las aplicaciones informáticas que son desarrolladas por sus propios programadores, pero además tiene que receptar e incorporar los proyectos desarrollados por terceros, como los proyectos implementados por los estudiantes de grado debido a materias de titulación o vinculación, y también los proyectos implementados por las unidades académicas de la IES o empresas externas.

Para empezar, se ha identificado que los programadores usan herramientas de codificación, librerías y tecnologías basándose en sus

conocimientos personales, y no se sigue un estándar en cuanto a la elección de los mismos. De igual manera, se ha identificado que los programadores canalizan sus esfuerzos en cumplir con la funcionalidad que las aplicaciones deben realizar, pero no se preocupan por crear un software que sea seguro y eficiente, ni mucho menos se incluyen pruebas de seguridad durante la fase de desarrollo.

En otras palabras, se evidencia la ausencia de protocolos o lineamientos a seguir para el desarrollo seguro de software, por lo que la DTI se encuentra propensa a sufrir ataques informáticos como la filtración de datos sensibles o ataques de denegación de servicios que afecten directamente la disponibilidad de las aplicaciones informáticas, inclusive puede dar paso a consecuencias devastadoras para la IES tanto en términos de daños financieros como de reputación.

Por otra parte, se ha detectado que el proceso de despliegue o pase a producción de las diversas aplicaciones informáticas se encuentra centralizado porque es realizado por una sola persona. La centralización del despliegue de aplicaciones hacia un solo desarrollador da como resultado problemas de disponibilidad y retrasos en los tiempos de entrega de las mejoras y actualizaciones de las aplicaciones informáticas.

1.3. Solución Propuesta

Se propone como solución a la problemática expresada anteriormente, la elaboración de un marco de seguridad que sirva de guía a la DTI para optimizar los procesos de desarrollo y despliegue de sus aplicaciones informáticas.

Para lograr este objetivo, se plantea utilizar las recomendaciones propuestas por la Organización Internacional de Normalización (ISO) como es la norma ISO/IEC 27002:2022, la cual es usada como un marco de referencia en la implementación de controles para el tratamiento de riesgos de la seguridad de la información [2]. El propósito de este proyecto se enfoca principalmente en los controles tecnológicos establecidos en dicha norma. Estos controles, en su mayoría de carácter preventivo, nos proporcionarán directrices para reducir la probabilidad de ser atacados por las vulnerabilidades de las aplicaciones informáticas, además de mitigar los riesgos asociados con posibles amenazas que podrían ser explotadas a través de estas vulnerabilidades.

Por otra parte, el Instituto Nacional de Estándares y Tecnología de EE.UU. (NIST) a través del Marco de Desarrollo de Software Seguro (SSDF), por sus siglas en inglés, nos proporciona recomendaciones para el proceso de desarrollo seguro de software que pueden ser

utilizadas en diferentes tipos de organizaciones. Esta guía de NIST puede ser usada como parte de las metodologías de desarrollo existentes que utiliza una organización, permitiéndoles producir software seguro con mínimas vulnerabilidades al momento de realizar el despliegue y prevenir vulnerabilidades similares a futuro, adicionalmente este marco de trabajo es aplicable en cualquier tipo de software independientemente de la tecnología o lenguaje de programación seleccionado [3].

Cabe recalcar que estos estándares internacionales no son absolutos, por este motivo se realizará una comparativa entre las especificaciones dispuestas y se definirán los controles y recomendaciones aplicables, que servirán de base para elaborar las políticas del marco de seguridad.

1.4. Objetivo General

Diseñar un marco de seguridad a partir de un análisis de riesgos de los activos de información para optimizar los procesos de desarrollo y despliegue de las aplicaciones informáticas que están bajo la responsabilidad de la Dirección de Tecnologías de la Información de una Institución de Educación Superior pública en Ecuador.

1.5. Objetivos Específicos

1) Determinar el contexto actual de la DTI en cuanto a los procesos de desarrollo y despliegue de aplicaciones informáticas.

2) Identificar los activos y riesgos de información asociados al proceso de desarrollo y despliegue de aplicaciones informáticas.
3) Establecer los controles necesarios dentro de las normativas internacionales ISO/IEC 27002:2022 y el marco de trabajo SSDF de NIST para mitigar los riesgos de los activos identificados.
4) Proponer un marco de seguridad basado en estándares internacionales para optimizar los procesos de desarrollo y despliegue de las aplicaciones informáticas.

1.6. Metodología

El presente trabajo es un estudio transversal de carácter descriptivo y de tipo no experimental basado en encuestas, y tiene como alcance el diseño de un marco de seguridad para la gestión de procesos de desarrollo y despliegues de aplicaciones, considerando los controles de la norma ISO/IEC 27002:2022 y el marco de trabajo SSDF de NIST.

Para ello, se realizará un proceso de levantamiento de información a través de la revisión de los actuales procesos que mantiene la DTI, además de cuestionarios en línea y entrevistas con los actores involucrados. La información proporcionada, en conjunto con la identificación de los activos, amenazas y riesgos de los sistemas de información permitirá definir el marco de seguridad que servirá como guía para la DTI.

En este trabajo se va a realizar un muestreo de tipo no probabilístico por conveniencia. La DTI se encuentra conformada por un conjunto de 50 personas, pero el Departamento de Desarrollo de Aplicaciones (DDA) está integrado aproximadamente por 30 personas, a quienes se consultarán sobre la situación en materia de seguridad de desarrollo de software.

Como instrumentos se usarán cuestionarios en línea y entrevistas. El cuestionario en línea estará compuesto de 50 preguntas aproximadamente y será elaborado con Microsoft Forms, mientras que las entrevistas serán presenciales en base a una ficha para profundizar aspectos de la encuesta. El cuestionario propuesto estará dirigido a los miembros del DDA y contendrá temas relacionados a los controles de la norma ISO/IEC 27002:2022, por ejemplo:

- Autenticación segura
- Codificación segura
- Ciclo de vida de desarrollo seguro
- Arquitectura de sistemas seguros y principios de ingeniería
- Pruebas de seguridad en el desarrollo y la aceptación
- Separación de los entornos de desarrollo, prueba y producción

Después se realizará un análisis cuantitativo con la información obtenida del cuestionario en línea, utilizando técnicas para identificar las

tendencias de los datos en materia de seguridad. También se realizará un análisis cualitativo proveniente de las entrevistas para conocer en más detalle el contexto sobre las prácticas de seguridad y los riesgos respecto a los procesos de desarrollo dentro de la DTI.

Con el análisis previo se pretende conocer cuáles son los retos que los programadores enfrentan para desarrollar una aplicación de software y los dolores que presentan para desplegar o realizar los pases a producción de las aplicaciones informáticas. Por otra parte, se desea conocer que tan informado está el personal sobre los controles establecidos por la norma ISO/IEC 27002:2022 y si es que actualmente se están aplicando.

Se espera que los resultados nos proporcionen la información necesaria para identificar de forma clara los posibles riesgos en materia de seguridad de la información. Además, los resultados encontrados servirán de base para el diseño de un marco integral de seguridad sobre los procesos de desarrollo y despliegue considerando los controles de la norma ISO/IEC 27002:2022 y las recomendaciones del marco de trabajo SSDF de NIST.

CAPÍTULO II

MARCO TEÓRICO

2.1. Marco de Seguridad

2.1.1. Seguridad de la Información

La seguridad de la información tiene como objetivo minimizar las consecuencias de los incidentes de seguridad de la información a través de la implementación de un conjunto de controles aplicables a los procesos de negocio de una organización. Estos controles deben ser especificados y revisados siempre que sea necesario para cumplir con los objetivos del negocio [4].

2.1.2. Pilares Fundamentales de la Seguridad de la Información

La confidencialidad, la disponibilidad y la integridad forman los pilares fundamentales de la seguridad de la información y su

aseguramiento tiene como propósito el éxito y continuidad de una organización [4].

Confidencialidad

La confidencialidad significa que la información no puede ser revelada a personas o recursos no autorizados para garantizar la privacidad de los datos [4].

Disponibilidad

La disponibilidad consiste en que la información siempre sea accesible en cualquier instante de tiempo [4].

Integridad

La integridad se refiere a que la información no debe ser alterada sin autorización para garantizar la exactitud de los datos [4].

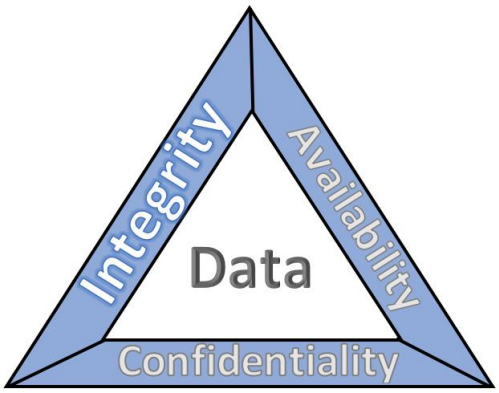

Figura 2.1: Pilares fundamentales de la información. Tomado de [3]

2.1.3. Amenazas y Vulnerabilidades de la Seguridad de la Información

2.1.3.1. Amenazas

De acuerdo con la norma ISO/IEC 27000 una amenaza es la causa potencial de un incidente no deseado, que puede afectar a un sistema o a la organización [5]. Dentro de las amenazas de seguridad de la información más comunes se obtiene [6]:

Denegación de servicio distribuidos (DDoS)

Este tipo de ataque consiste en sobrecargar la aplicación de una organización, lo que provoca un bloqueo o denegación de servicio a los usuarios que tratan de acceder a dicha aplicación, interrumpiendo la disponibilidad del mismo.

Phishing

Este ataque tiene como objetivo engañar a los usuarios para que compartan información confidencial como números de tarjeta y contraseñas. En el phishing se envían mensajes fraudulentos ya sea por correo, por mensaje de texto u otro canal de comunicación.

Ransomware

El ransomware es un ataque extorsivo que cifra la información de los dispositivos de un usuario u organización impidiendo el acceso a los mismos. Para restaurar la información se exige el pago de un rescate por lo general en criptomonedas.

Hombre en el medio

En este tipo de ataque el atacante interrumpe la transferencia de datos entre dos partes haciéndose pasar por un usuario autorizado con el fin de monitorear, interceptar o alterar la información que se está transmitiendo.

2.1.3.2. Vulnerabilidades

Una vulnerabilidad es la debilidad de un activo o control que puede ser explotado por uno o más amenazas [4], entendiéndose como activo a los sistemas informáticos que gestiona o controla una organización. De acuerdo con OWASP [7] las 10 principales vulnerabilidades de seguridad presentes en las aplicaciones web con relación al año 2021 son las siguientes:

Pérdida de Control de Acceso: Un control de acceso que no funciona permite la divulgación no autorizada de información.

Fallas criptográficas: Son fallos relacionados con la encriptación de datos, lo que puede conducir a la exposición de datos confidenciales.

Inyección: Un ataque de inyección de código consiste en insertar código malicioso en una aplicación.

Diseño inseguro: Errores en el diseño de una aplicación, lo que puede conllevar a tener componentes con vulnerabilidades.

Configuración de seguridad incorrecta: Debilidad por falta de configuración de parámetros de seguridad o usar valores por defecto.

Componentes vulnerables y desactualizados: Los componentes obsoletos presentes en las aplicaciones pueden ser aprovechados por los atacantes para orquestar un incidente de seguridad.

Fallas de identificación y autenticación: Es una debilidad que se presenta cuando no se valida correctamente la identidad de un usuario.

Fallas en el software y en la integridad de los datos: Son fallos relacionados con la actualización de software, datos críticos y canalizaciones de CI/CD utilizadas sin verificar la integridad.

Fallas en el registro y monitoreo: El monitoreo y registro son actividades esenciales que deben ser realizadas en las aplicaciones, para detectar problemas de seguridad.

Falsificación de solicitudes del lado del servidor: Ocurre cuando una aplicación busca un recurso remoto sin validar la URL proporcionada por el usuario, dando paso a que un atacante obligue a la aplicación enviar una solicitud a un destino inesperado.

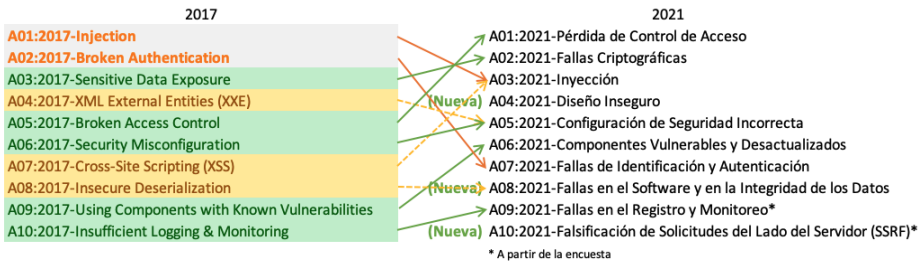

Figura 2.2: Top 10 de vulnerabilidades del año 2021 según OWASP. Tomado de [8]

2.1.4. Metodología de Gestión de Riesgos de la Seguridad de la Información

De acuerdo con MAGERIT, el riesgo es la estimación del grado de exposición a que una amenaza se materialice sobre uno o más activos causando daños o perjuicios a la organización [9]. Los riesgos se originan por distintas causas, como los desastres naturales, accidentes o amenazas de ciberseguridad. Por tanto, se debe gestionar los riesgos de una manera adecuada.

Gestión de Riesgos

La gestión de riesgos es un proceso que permite identificar, analizar, evaluar y tratar los riesgos asociados a la seguridad de la información, mediante los controles establecidos en una organización [10]. La metodología de la gestión de riesgos está compuesta de 4 etapas:

Identificación del riesgo

Para poder identificar los riesgos se necesita identificar 3 elementos: los activos de información, las amenazas, los controles existentes.

Análisis del riesgo

El análisis del riesgo consiste en determinar la probabilidad, el impacto y el nivel de riesgo al que se encuentra expuesto.

Evaluación del riesgo

La evaluación del riesgo implica relacionar los activos de información con los pilares de la seguridad de la información.

Tratamiento del riesgo

Finalmente, para cada riesgo identificado se requiere seleccionar la opción de tratamiento más adecuada y aplicarla. Las opciones de tratamiento de riesgos son: Mitigar, Aceptar, Transferir y Evitar (MATE).

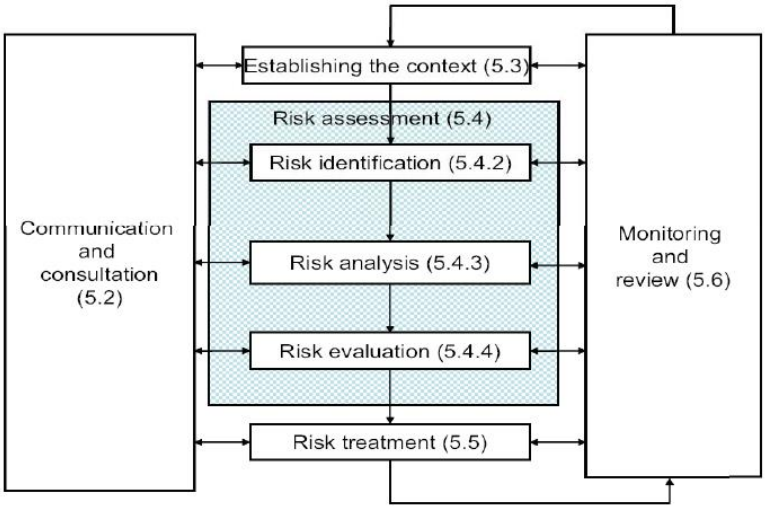

Figura 2.3: Metodología de gestión de riesgos. Tomado de [10]

2.2. Normativas Internacionales

2.2.1. ISO/IEC 27001

Esta norma proporciona los requisitos para el establecimiento, implementación, mantenimiento y mejora continua de un Sistema

de Gestión de la Seguridad de Información (SGSI). También puede ser usada para evaluar la capacidad de la organización para cumplir con sus propios requisitos de seguridad [11].

2.2.2. ISO/IEC 27002

Esta norma es usada como referencia para determinar e implementar controles para el tratamiento de riesgo para la seguridad de información. Definiendo como control una medida que modifica o mantiene el riesgo de un incidente de información, personal o infraestructura física [2].

2.2.3. ISO/IEC 31000

Esta norma proporciona los principios y directrices de carácter genérico sobre la gestión de riesgos. Se puede aplicar a cualquier tipo de riesgo, cualquiera que sea su naturaleza y puede ser aplicada por cualquier tipo de institución ya sea pública o privada [10].

2.2.4. NIST

La publicación 800-218 del Instituto Nacional de Estándares y Tecnología de EE.UU. (NIST) hace referencia a un Marco de Desarrollo de Software Seguro (SSDF) que son un conjunto de recomendaciones para mitigar los riesgos de las vulnerabilidades del software [12].

2.3. Procesos de Desarrollo y Despliegue

2.3.1. Ciclo de Vida del Desarrollo de Software y su Importancia

El ciclo de vida del desarrollo de software (SDLC) es un método que incluye un conjunto de actividades, desde la toma de requerimientos con el cliente hasta el desarrollo y soporte de un sistema informático [13].

El desarrollo de software puede ser complejo de gestionar debido a los requisitos cambiantes y a los avances tecnológicos. Algunas ventajas del SDLC son [8]:

- Mayor visibilidad del proceso de desarrollo para todas las partes interesadas.
- Una estimación, planificación y programación eficiente.
- Mejoras en la administración de riesgos y estimación de costos.
- Entregas de software sistemáticas y mayor satisfacción de los clientes.

2.3.2. Etapas y Modelos del Ciclo de Vida del Desarrollo de Software

2.3.2.1. Etapas

El ciclo de vida del desarrollo de software está compuesto de varias etapas [8]:

Planificación

En esta fase el equipo de desarrollo levanta toda la información necesaria de las partes interesadas, como clientes o directivos. Se estiman los costos, se asignan recursos, se definen los requerimientos del software y el plan de ejecución del proyecto.

Diseño

En esta fase se analizan los requerimientos y se identifican las mejores soluciones para crear el software. Se crean prototipos para modelar el funcionamiento de la aplicación.

Implementación

En la fase de implementación o desarrollo se codifica el software como tal, en base a los lenguajes de programación previamente seleccionados.

Pruebas

Esta fase consiste en realizar la mayor cantidad de pruebas, ya sean manuales o automáticas, para encontrar errores y corregirlos para asegurar el funcionamiento del software.

Despliegue

Esta fase significa que se debe pasar a un entorno de producción el software que se ha desarrollado para que sea accesible por las partes interesadas.

Mantenimiento

Dentro de la fase de mantenimiento se atienden nuevos requerimientos de los clientes, se corrigen errores no descubiertos con el propósito de mejorar el software existente.

2.3.2.2. Modelos

Existen varios modelos del ciclo de vida del desarrollo de software para gestionar proyectos, entre los más comunes se presentan los siguientes [8]:

Cascada

El modelo en cascada modelo propone que todas las etapas del SDLC se implementen secuencialmente, de

modo que la nueva etapa dependa de los resultados de la etapa anterior.

Iterativo

Este modelo corresponde a la iteración de varios ciclos de vida en cascada. Se produce una nueva versión del software al final de cada iteración.

Espiral

El modelo en espiral combina los ciclos del modelo iterativo con el flujo secuencial del modelo en cascada. Con este modelo se busca la actualización y mejoras graduales del software.

Ágil

Un modelo ágil es un modelo flexible donde las etapas del SDLC se dividen en varios ciclos de desarrollo. Se itera rápidamente a través de las etapas y se obtienen pequeñas partes del software en cada iteración, aumentando la satisfacción de los clientes.

CAPÍTULO III

CONTEXTO DE LA ORGANIZACIÓN

3.1. Revisión de Procesos Vigentes de la Organización

Se ha realizado una revisión a través de los diversos portales web que posee la Institución de Educación Superior (IES) sujeta a este estudio y se ha encontrado que existen varios procesos documentados y modelados mediante diagramas de flujo que involucran a la Dirección de Tecnologías de la Información (DTI) en diferentes aspectos.

Si bien es cierto que contamos con esta información, este trabajo se enfoca en los procesos que están relacionados con el proceso interno de desarrollo de aplicaciones. Estos procesos son los siguientes:

Gestión de Requerimientos

Este proceso sirve para gestionar los requerimientos informáticos de las unidades académicas y administrativas de la IES, es decir se define la

viabilidad de la implementación, la evaluación de los riesgos funcionales y técnicos, la estimación de tiempo y costos, así como la asignación de recursos necesarios.

Gestión de Proyectos de Desarrollo de Aplicaciones

Este proceso tiene como objetivo determinar los lineamientos para gestionar los proyectos de desarrollo de aplicaciones que recepta la DTI, con el fin de satisfacer las necesidades de todas las partes interesadas.

Administración de Base de Datos

Este proceso tiene como objetivo definir qué datos deben almacenarse en la base de datos y establecer las directrices para manejar los datos una vez almacenados.

Gestión de Activos de TI

Este proceso define los lineamientos para gestionar los activos de la tecnología de la información, con el fin de contar con un registro de los activos y poder dar seguimiento para garantizar las condiciones de funcionamiento.

Gestión de Vulnerabilidades y Amenazas de TI

Este proceso determina los lineamientos para gestionar las vulnerabilidades y amenazas de TI a través de medidas preventivas, detección y correctivas.

Gestión de Riesgos de TI

Este proceso tiene como objetivo determinar los lineamientos necesarios para identificar y evaluar los riesgos de la tecnología de la información, de acuerdo con su impacto y probabilidad de ocurrencia. Los riesgos identificados deben ser tratados con controles y planes de acción que permitan reducir su impacto ante posibles incidentes.

3.2. Incidentes de Seguridad Ocurridos en la Organización

De acuerdo con relatos por parte de los empleados de la DTI, se han registrado incidentes que han afectado la confidencialidad, integridad y disponibilidad de los sistemas de información que ellos administran. Algunos ejemplos de estos incidentes son los siguientes:

- Estudiantes y personal administrativo de la IES han sido víctimas de ataques de ingeniería social y phishing por el envío de correos electrónicos maliciosos dentro de los correos institucionales.
- Los errores de configuración a nivel de aplicación para soportar una alta demanda de usuarios concurrentes en la plataforma de gestión de aprendizaje institucional provocaron problemas de disponibilidad impidiendo que los estudiantes puedan rendir sus evaluaciones en horas pico.
- El proceso de registro de los estudiantes en las materias planificadas del periodo académico que se realiza a través del

sistema académico se vio afectado un tiempo considerable debido a cambios de último momento en la base de datos y errores de configuración.

- Un sitio web elaborado con Wordpress fue vulnerado por un plugin que ya no contaba con soporte, específicamente se logró inyectar código malicioso en el código fuente de la aplicación, lo que ocasionó que el sitio web colapse definitivamente.
- Un sitio web construido con Drupal se vio interrumpido por utilizar módulos desactualizados que ya no eran compatibles con la versión del lenguaje de programación instalado inicialmente.
- Una plataforma de consulta bibliográfica quedó fuera de servicio por una ineficiente asignación de recursos en el servidor en cuanto a memoria y espacio en disco.
- Se descubrió también que un equipo fue vulnerado a través de una aplicación externa para acceso remoto.
- La DTI ha enfrentado un problema recurrente relacionado con el abuso de privilegios ya que, en varias ocasiones, empleados han utilizado de manera indebida su acceso al sistema para modificar directamente las calificaciones de estudiantes.

De los sucesos mencionados previamente se puede obtener la siguiente clasificación de incidentes de seguridad:

- Ingeniería social
- Phishing
- Errores de configuración
- Cambios en la base de datos
- Fuga de datos
- Intrusiones y Ataques Internos/Externos
- Vulnerabilidades de software

3.3. Levantamiento de Información a través de Entrevistas y Cuestionarios

Continuando con el proceso de recopilación de información, se ha diseñado un cuestionario en línea compuesto por 49 preguntas, las cuales se encuentran alineadas con los controles organizacionales y tecnológicos establecidos en la norma ISO/IEC 27002:2022. Este cuestionario específicamente se ha dirigido a los desarrolladores de la organización con el propósito de obtener una evaluación detallada sobre la seguridad en el desarrollo de software.

Además, para profundizar en los temas abordados en la encuesta, se ha llevado a cabo entrevistas con los jefes de cada departamento de la DTI. Estas entrevistas buscan ahondar en aspectos cruciales como las medidas de seguridad implementadas en la DTI y la gestión de incidentes de seguridad.

Este enfoque integral, que combina la perspectiva de los desarrolladores con la visión de los jefes, nos proporciona una evaluación completa y detallada del estado actual de la seguridad de desarrollo de software en la organización.

3.4. Análisis de los Resultados

El cuestionario en línea dirigido a los desarrolladores de la DTI obtuvo la participación de 20 personas, quienes generosamente compartieron sus valiosas perspectivas. Este cuestionario abarca un amplio espectro de temas organizados en 7 secciones con el fin de evaluar el conocimiento de los desarrolladores sobre las mejores prácticas de seguridad de la información. A continuación, se describen los resultados para cada sección:

Política de seguridad de la información

La mayoría de los desarrolladores considera que la seguridad de la información es una prioridad en la organización y también están al tanto de la existencia de una política de seguridad, aunque hay un porcentaje significativo que aún no está informado. Además, tienen el conocimiento sobre a quién informar en caso de incidentes de seguridad.

Por otra parte, la mayoría de los encuestados indicó que no se proporciona formación periódica sobre seguridad de la información y

que no están seguros si se efectúa regularmente una evaluación de riesgos de seguridad de la información.

Roles, responsabilidades y derechos de propiedad intelectual

Aunque la mayoría de encuestados reconoce que existen roles y responsabilidades claramente definidos para la seguridad de la información, un número significativo de encuestados no está seguro sobre su existencia. La falta de certeza sobre la revisión periódica de los roles y responsabilidades destaca la importancia de mejorar la transparencia en la gestión de la seguridad de la información.

En cuanto a los derechos de propiedad intelectual, hay un fuerte reconocimiento de la importancia de la propiedad intelectual como un activo crítico para la organización, pero hay incertidumbre sobre la existencia de políticas específicas y la clasificación adecuada de la propiedad intelectual.

Derechos de acceso privilegiados

El 90% de los encuestados indica que se otorgan derechos de acceso privilegiados a los usuarios según los procesos o sistemas de información que tienen a cargo, pero una minoría de encuestados afirma que se supervisan las actividades de dichos usuarios. Finalmente, solo el 30% de los encuestados está seguro de que se ha

definido un procedimiento para la revocación de derechos de acceso privilegiados cuando ya no son necesarios.

Restricción de acceso a la información y Acceso al código fuente

El 85% de los encuestados afirma que se controlan los permisos de acceso de los usuarios, incluyendo permisos de lectura, escritura y ejecución, así como la gran mayoría de los empleados indica que se limita la información contenida en las aplicaciones de acuerdo con el tipo de usuario. Por otra parte, el 60% de los encuestados opina que se proporcionan controles de acceso físico o lógico para aislar las aplicaciones o sistemas sensibles.

Con respecto a la sección de acceso al código fuente, una alta cantidad de encuestados manifiesta que se otorgan permisos de lectura y escritura al código fuente según las necesidades de la organización, y que la forma de acceder al repositorio del código fuente es a través de herramientas de desarrollo.

Seguridad de autenticación y Enmascaramiento de datos

El 55% de los encuestados respondió que implementan medidas para ataques de fuerza bruta en los sistemas de información que requieren autenticación, esto incluye prácticas como el uso de captcha y restablecimiento de contraseña, incluso la mayoría de los desarrolladores optan por restringir o cerrar la sesión después de un

tiempo de inactividad en los sistemas, una práctica común para mitigar riesgos asociados con la inactividad de la sesión, sin embargo no están seguros de que se realice un registro de los intentos de inicio de sesión ya sean exitosos o fallidos.

En cuanto al enmascaramiento de datos, a pesar de que una parte significativa de los desarrolladores afirma aplicar técnicas de enmascaramiento de datos, hay espacio para mejorar la implementación de estas prácticas. La organización podría beneficiarse de iniciativas que refuercen la importancia del enmascaramiento de datos para garantizar la protección adecuada de la información sensible.

Logging, Criptografía y Ciclo de vida de desarrollo seguro

Existe un nivel de conciencia razonable por parte de los desarrolladores sobre la información que se almacena en los logs y las medidas de protección contra manipulaciones no autorizadas como la eliminación o alteración de registros. A pesar de esto, hay oportunidades para mejorar el conocimiento y la práctica relacionada con el monitoreo de logs en busca de actividades inusuales o amenazas.

En el área de uso de algoritmos criptográficos, de manera general, se observa un desconocimiento de los encuestados para determinar qué información debe ser cifrada y cuáles son los algoritmos de cifrado que

se aplican dentro de la organización. De similar forma, los empleados piensan que no se recibe una capacitación regular en técnicas criptográficas, lo que sugiere una oportunidad para mejorar la formación en este aspecto crítico de seguridad.

Finalmente, aunque los desarrolladores se sienten capacitados para encontrar y corregir vulnerabilidades presentes en el código, es necesario que se definan estándares y directrices para el desarrollo seguro de software en la organización, ya que solo el 30% de los encuestados manifiesta que se realizan evaluaciones de seguridad antes de implementar nuevas funcionalidades o versiones de software, y un porcentaje menor afirma que se llevan a cabo pruebas de seguridad periódicas desde las fases iniciales hasta las fases finales del ciclo de vida del desarrollo.

Arquitectura de sistemas seguros, Código seguro y Separación de ambientes

El 50% de los encuestados respondió conocer que las aplicaciones y sistemas de información de la organización se encuentran debidamente hardenizados. Este resultado sugiere una división entre aquellos que están al tanto de las prácticas de hardening y aquellos que no lo están.

El 90% de los empleados utilizan un IDE como herramienta de desarrollo para forzar la creación de código seguro, ya que estas

herramientas son actualizadas constantemente. No obstante, solo un porcentaje menor de encuestados elimina fallos de programación que podrían provocar vulnerabilidades, tampoco se documenta como fueron mitigados los errores comunes de programación, inclusive no se controla totalmente prácticas inseguras en el diseño de software como el uso de servicios web inseguros. En gran parte, esto es debido a que solo el 35% de los desarrolladores aplican estándares de desarrollo seguro dependiendo del lenguaje programación utilizado, aunque los encuestados si se toman la molestia de revisar que las librerías empleadas en el código tengan soporte y provengan de fuentes confiables.

En cuanto a la separación de ambientes, todos los desarrolladores respondieron que se separa adecuadamente los ambientes de desarrollo y producción, y la gran mayoría afirma que se realizan respaldos de dichos ambientes, poniendo de manifiesto una buena práctica de seguridad para garantizar la disponibilidad y recuperación en caso de eventos no deseados. De similar forma, un alto porcentaje de los encuestados asegura que los cambios requeridos en los sistemas primero se implementan en un entorno de prueba antes de aplicarlos a los sistemas de producción, y que se utilizan las mismas herramientas y configuraciones en los entornos de desarrollo y producción. Como consecuencia de esto, el 90% de los desarrolladores presenta

apropiadamente la información de cada ambiente, lo que podría ayudar a evitar errores humanos al transferir datos entre los diferentes ambientes.

Complementado los resultados del cuestionario en línea con las entrevistas realizadas a los jefes de departamento, un punto clave a denotar es que la DTI no cuenta con una política de seguridad de la información formalmente establecida. Sin embargo, se han modelado procesos relacionados con la seguridad de la información y se están definiendo algunas regulaciones, como la protección de datos personales y política de cookies.

Además, se pudo constatar que, gracias al conocimiento empírico adquirido por el personal de la DTI se aplican ciertos controles recomendados por la norma ISO/IEC 27002:2022, tales como la asignación de roles y responsabilidades, el uso de herramientas de desarrollo seguro, el registro de logs en aplicaciones críticas y separación de ambientes de trabajo para prevenir incidentes.

CAPÍTULO IV

ACTIVOS Y RIESGOS DE INFORMACIÓN

4.1. Identificación de Activos

Los activos de información de la Dirección de Tecnologías de la Información (DTI) son todos aquellos recursos que tienen un valor para la organización y que deben ser protegidos. Estos activos pueden ser tangibles o intangibles, y están relacionados con el desarrollo, el mantenimiento o la operación de los sistemas de información.

El manejo adecuado de los activos de información es esencial en el Departamento de Desarrollo de Aplicaciones (DDA) para garantizar la seguridad, integridad y disponibilidad de los datos, por tanto, se propone categorizar los activos de la siguiente manera:

Tabla 1: Tipos de activo de información

Tipo de activo	Código
Datos/información	D
Hardware	HW
Software	SW
Personas	P

Los hallazgos derivados del análisis realizado en el capítulo previo nos permiten identificar los activos de información asociados al DDA. A continuación, se presenta la clasificación de los activos de información sujetos al presente estudio.

Tabla 2: Clasificación de los activos de información

Código activo	Nombre activo	Descripción
P01	Programadores	Personas encargadas de desarrollar las aplicaciones o sistemas de información.
SW01	Código fuente	Líneas de código de las aplicaciones.
SW02	Aplicaciones	Aplicaciones o sistemas de información que se encuentran en producción.
SW03	Base de datos	Datos de producción, copias de respaldo, esquemas de base de datos.
SW04	Frameworks y librerías	Frameworks utilizados en el desarrollo, librerías o componentes de terceros.
SW05	Logs	Registros de accesos, registros de errores o incidentes.
SW06	Plataformas de desarrollo	Herramientas que utilizan los programadores para crear software.
SW07	Licencias de software	Licencias para utilizar las herramientas de desarrollo.
HW01	Servidores	Servidor de aplicación, servidor de base de datos, etc.

HW02	Equipos de Computo	Laptops, Computadoras de escritorio, Accesorios de computación.
D01	Documentación	Documentación de manuales de usuario, manuales de instalación, procedimientos o procesos relacionados al ciclo de vida de desarrollo de software.

4.2. Identificación de Amenazas y Vulnerabilidades

4.2.1. Identificación de Amenazas

El desarrollo de software enfrenta diversas amenazas que pueden comprometer la seguridad, la integridad y la disponibilidad de los sistemas y datos. Basándonos en el origen de la amenaza, se pueden clasificar de la siguiente manera: naturales, industriales, errores o fallos no intencionados y ataques intencionados.

En la tabla 3, se proporciona un listado de posibles amenazas que podrían afectar las actividades vinculadas al desarrollo de software:

Tabla 3: Identificación de amenazas

Tipo de amenaza	Nombre de amenaza
Natural	Fuego Daños por agua Desastres naturales (tormentas eléctricas, rayos, sismos, etc.)
Industrial	Corte del suministro eléctrico Condiciones inadecuadas de temperatura o humedad Fallo de servicios de comunicaciones

	Fallo de funcionamiento de hardware
Errores o fallos no intencionados	Incompetencia Incumplimiento de políticas de seguridad Errores de configuración de seguridad
Ataques intencionados	Malware Ataques de fuerza bruta Inyección de código Phishing Ransomware Acceso no autorizado Alteración de información Fuga de información Robo de información Falsificación Extorsión Manipulación de programas Manipulación de logs

4.2.2. Identificación de Vulnerabilidades

Por otra parte, las vulnerabilidades que afectan el desarrollo de software son aquellas que pueden ser explotadas por un atacante para comprometer la seguridad de un sistema o aplicación. Estas vulnerabilidades pueden estar presentes en cualquier etapa del ciclo de vida de desarrollo del software, desde el diseño hasta la implementación y la operación.

Tomando como referencia la encuesta realizada en el capítulo anterior, se ha identificado las siguientes vulnerabilidades según el activo de información:

Tabla 4: Identificación de vulnerabilidades

Código activo	Nombre activo	Vulnerabilidad
D01	Documentación	No documentar las reglas de implementación y autorización durante el proceso de desarrollo
HW01	Servidores	Falta de hardening de seguridad
		Falta de mantenimiento preventivo de equipos
		Falta de uso de certificados SSL para activar protocolo HTTPS
		Mala asignación de recursos por usuario o servicio
		Falta de mantenimiento preventivo de equipos
P01	Programadores	Apertura de archivos maliciosos
		Errores de codificación
		Falta de capacitación y conocimientos de programación
		Uso de contraseñas débiles o predeterminadas
SW01	Código fuente	No separar las configuraciones de los ambientes de desarrollo y producción
		No usar patrones de diseño de software
		No usar repositorios para versionamiento del código fuente
		Uso de componentes o librerías con vulnerabilidades conocidas o sin mantenimiento
		No usar herramientas de análisis de código estático o dinámico
		Uso de algoritmos criptográficos obsoletos o débiles
SW02	Aplicaciones	Manipulación de metadatos como JWT o cookies
		No contar con ambientes de desarrollo y producción
		Asignación de privilegios incorrecta
		Configuración incorrecta de las cabeceras HTTP
		Elevación de privilegios
		Exponer APIs inseguros
		No cifrar datos sensibles
		No expirar las sesiones
		No implementar autenticación multifactor
		No realizar pruebas unitarias y de integración
		No usar captcha en formularios
		No utilizar ORM para acceder a la base de datos
		No validar la entrada de datos
		Permitir múltiples registros fallidos de autenticación

		Presentar mensajes de error detallados
		Usar valores por defecto de las configuraciones de seguridad
SW03	Base de datos	Falta de respaldos de la base de datos
		No contar con ambientes de desarrollo y producción
		Almacenamiento de contraseñas en texto plano
		Asignación de privilegios incorrecta
		Usar valores por defecto de las configuraciones de seguridad
SW04	Frameworks y librerías	Uso de algoritmos criptográficos obsoletos o débiles
SW05	Logs	Exponer datos sensibles en los logs
		No generar alertas de actividad inusual
		No registrar errores de las aplicaciones
		No registrar inicios de sesión exitosos y fallidos
		No respaldar periódicamente los logs

4.3. Identificación y Análisis de Riesgos

4.3.1. Identificación de Riesgos

La identificación de riesgos es un proceso que requiere de la identificación de los activos de información, las amenazas y vulnerabilidades. En las 2 secciones previas de este capítulo se definieron dichos elementos.

4.3.2. Análisis de Riesgos

Para realizar un análisis de los riesgos asociados a las vulnerabilidades previamente descritas, primero hay que determinar la probabilidad y el impacto al que se encuentran expuestos.

Evaluación de la Probabilidad

Se define la matriz de probabilidad con una escala de 4 niveles.

Tabla 5: Matriz de probabilidad

PROBABILIDAD		
Escala	**Valor**	**Descripción**
Constante	4	Altamente probable, ocurrencia frecuente
Moderado	3	Probabilidad significativa, ocurrencia común
Ocasional	2	Probabilidad casual, ocurrencia repentina
Improbable	1	Ocurrencia altamente improbable

Evaluación del Impacto

Se define la matriz de impacto con una escala de 4 niveles.

Tabla 6: Matriz de impacto

IMPACTO		
Escala	**Valor**	**Descripción**
Muy alto	4	Interrumpe totalmente la disponibilidad de las aplicaciones. Afecta en gran escala la reputación de la IES. Perdida de dinero por incidente. Alteración de información.
Alto	3	Interrumpe parcialmente la disponibilidad de las aplicaciones. Afecta de forma moderada la reputación de la IES. Incumplimiento regulatorio. Divulgación de información.
Medio	2	Afecta de forma moderada a un estudiante, profesor, personal administrativo o a la DTI. Reputación afectada pero recuperable.
Bajo	1	No hay impacto financiero, regulatorio o de reputación. No detiene las operaciones.

4.3.3. Mapa de Calor

En este análisis de riesgos se utiliza el siguiente esquema de mapa de calor con los valores asignados que se visualizan en la tabla 7:

Tabla 7: Mapa de calor

	IMPACTO			
PROBABILIDAD	**Bajo (1)**	**Medio (2)**	**Alto (3)**	**Muy alto (4)**
Constante (4)	4	8	12	16
Moderado (3)	3	6	9	12
Ocasional (2)	2	4	6	8
Improbable (1)	1	2	3	4

Valoración del Riesgo

El nivel de exposición al riesgo es determinado por la valoración del impacto y la probabilidad. Para este trabajo se definen 4 niveles de riesgos con las siguientes ponderaciones:

Tabla 8: Valoración del riesgo

Nivel	Calificación
Grave	12 a 16
Importante	8 a 11
Medio	4 a 7
Bajo	1 a 3

4.3.4. Asignación de Impacto y Probabilidad a los Riesgos

La siguiente tabla muestra la valoración de los riesgos según la ponderación de la probabilidad y el impacto de las vulnerabilidades en los diferentes activos de información.

Tabla 9: Asignación de Impacto y Probabilidad

Código activo	Vulnerabilidad	Probabilidad	Impacto	Riesgo
D01	No documentar las reglas de implementación y autorización durante el proceso de desarrollo	3	3	9
HW01	Falta de hardening de seguridad	1	4	4
HW01	Falta de mantenimiento preventivo de equipos	2	4	8
HW01	Falta de uso de certificados SSL para activar protocolo HTTPS	2	2	4
HW01	Mala asignación de recursos por usuario o servicio	2	4	8
HW02	Falta de mantenimiento preventivo de equipos	2	3	6
P01	Apertura de archivos maliciosos	2	3	6
P01	Errores de codificación	3	4	12
P01	Falta de capacitación y conocimientos de programación	2	2	4
P01	Uso de contraseñas débiles o predeterminadas	3	3	9
SW01	No separar las configuraciones de los ambientes de desarrollo y producción	2	4	8
SW01	No usar patrones de diseño de software	3	1	3
SW01	No usar repositorios para versionamiento del código fuente	1	4	4
SW01	Uso de componentes o librerías con vulnerabilidades conocidas o sin mantenimiento	2	2	4
SW01	No usar herramientas de análisis de código estático o dinámico	3	3	9

SW01	Uso de algoritmos criptográficos obsoletos o débiles	2	3	6
SW02	Manipulación de metadatos como JWT o cookies	1	4	4
SW02	No contar con ambientes de desarrollo y producción	1	3	3
SW02	Asignación de privilegios incorrecta	1	4	4
SW02	Configuración incorrecta de las cabeceras HTTP	2	2	4
SW02	Elevación de privilegios	1	4	4
SW02	Exponer APIs inseguros	2	4	8
SW02	No cifrar datos sensibles	1	3	3
SW02	No expirar las sesiones	2	4	8
SW02	No implementar autenticación multifactor	4	3	12
SW02	No realizar pruebas unitarias y de integración	2	3	6
SW02	No usar captcha en formularios	2	3	6
SW02	No utilizar ORM para acceder a la base de datos	1	3	3
SW02	No validar la entrada de datos	2	3	6
SW02	Permitir múltiples registros fallidos de autenticación	2	3	6
SW02	Presentar mensajes de error detallados	2	2	4
SW02	Usar valores por defecto de las configuraciones de seguridad	2	4	8
SW03	Falta de respaldos de la base de datos	1	4	4
SW03	No contar con ambientes de desarrollo y producción	1	4	4
SW03	Almacenamiento de contraseñas en texto plano	2	4	8
SW03	Asignación de privilegios incorrecta	1	4	4
SW03	Usar valores por defecto de las configuraciones de seguridad	2	4	8
SW04	Uso de algoritmos criptográficos obsoletos o débiles	1	3	3
SW05	Exponer datos sensibles en los logs	1	4	4

SW05	No generar alertas de actividad inusual	3	3	9
SW05	No registrar errores de las aplicaciones	2	3	6
SW05	No registrar inicios de sesión exitosos y fallidos	2	3	6
SW05	No respaldar periódicamente los logs	2	4	8

CAPÍTULO V

SELECCIÓN DE CONTROLES Y RECOMENDACIONES PARA MITIGAR RIESGOS

5.1. Controles Aplicables de la Norma ISO/IEC 27002:2022

Para el desarrollo del presente trabajo de titulación estamos usando como referencia la norma ISO/IEC 27002 versión 2022 que es un estándar internacional que proporciona las directrices sobre la implementación de controles de seguridad de la información. Dicha norma consta de 93 controles agrupados en 4 clausulas: Controles Organizacionales, Controles de Personas, Controles Físicos y Controles Tecnológicos.

A continuación, se presenta una tabla que incluye los controles que se han seleccionado debido a que son aplicables en la elaboración del

marco de seguridad para la optimización de los procesos de desarrollo y despliegue.

Tabla 10: Descripción de controles aplicables de la norma ISO/IEC 27002:2022

Control	Nombre del control	Descripción del control
5.1	Políticas de seguridad de la información	Establecer políticas para asegurar que la dirección de la gestión de la seguridad de la información sea adecuada y esté alineada con los requisitos comerciales y legales correspondientes.
5.2	Roles y responsabilidades de seguridad de la información	Definir roles y responsabilidades que permitan crear una estructura clara y aceptada para operar y gestionar la seguridad de la información dentro de la organización.
5.15	Control de Acceso	Controlar el acceso físico y lógico a la información para garantizar el acceso autorizado y prevenir el no autorizado a la información y otros activos asociados.
5.17	Información de autenticación	Proteger la información, como contraseñas o tokens, usada para autenticar y controlar el acceso a los sistemas y servicios con el objetivo de prevenir el acceso no autorizado.
5.32	Derechos de propiedad intelectual	Cumplir con las obligaciones legales, normativas y contractuales en relación con los derechos de propiedad intelectual y la utilización de productos patentados.
6.3	Sensibilización, educación y formación en materia de seguridad de la información	Informar y formar a todos los empleados acerca de sus responsabilidades en materia de seguridad de la información.
7.13	Mantenimiento de equipos	Prevenir perdidas, daños, hurtos o comprometer la información y otros activos asociados e interrumpir las operaciones de la organización

		causado por la carencia de mantenimientos.
8.2	Derechos de acceso privilegiados	Garantizar que solo los usuarios autorizados, los componentes y servicios de software se proporcionen con derechos de acceso privilegiados.
8.4	Acceso al código fuente	Prevenir la introducción de funciones no autorizadas, cambios involuntarios o maliciosos para mantener la confidencialidad de la propiedad intelectual.
8.5	Autenticación segura	Garantizar que un usuario o una entidad esté autenticado de forma segura, cuando acceda a sistemas o aplicaciones.
8.6	Gestión de la capacidad	Asegurar que los recursos de procesamiento de la información tengan la capacidad adecuada para cumplir con los requisitos actuales y futuros de una manera eficaz y eficiente.
8.7	Protección contra el malware	Implementar defensas eficaces para prevenir, detectar y eliminar el software malicioso en toda la organización.
8.9	Gestión de la configuración	Mantener una gestión adecuada de las configuraciones de hardware y software de los sistemas de información dentro de una organización.
8.11	Enmascaramiento de datos	Limitar la exposición de datos confidenciales para cumplir con los requisitos legales, estatutarios, reglamentarios y contractuales.
8.13	Información de respaldo	Asegurar que las copias de seguridad de los datos esenciales de la organización se realizan de manera regular, segura y eficaz, para garantizar la disponibilidad de la información en caso de incidentes.

8.15	Registro	Registrar eventos, preservar evidencia y asegurar la integridad de los registros como medidas para prevenir el acceso no autorizado y facilitar la identificación de eventos de seguridad que puedan dar lugar a incidentes.
8.16	Actividades de monitoreo	Supervisar las actividades relacionadas con la seguridad de la información con el fin de detectar eventos y anomalías que puedan indicar problemas de seguridad.
8.20	Seguridad de las redes	Proteger la información que se transmite a través de las redes para salvaguardar la información en todas las etapas de su transmisión, desde el origen hasta el destino.
8.24	Uso de criptografía	Utilizar la criptografía de manera adecuada y efectiva para salvaguardar aspectos clave de la información, cumpliendo con requisitos de seguridad, así como considerando las obligaciones legales y contractuales asociadas a la criptografía.
8.25	Ciclo de vida de desarrollo seguro	Integrar la seguridad de la información en el proceso de desarrollo de software, asegurando que esté presente desde las fases iniciales hasta la implementación final.
8.26	Requisitos de seguridad de aplicaciones	Asegurar todos los requisitos de seguridad de información sean identificados y dirigido cuando la aplicación se encuentre en desarrollo o sea adquirida.
8.27	Arquitectura de sistemas seguros y principios de ingeniería	Asegurar que los sistemas de información sean diseñados, implementados y operados de manera segura en todas las etapas del ciclo de vida del desarrollo de software.

8.28	Codificación segura	Escribir software de manera segura con el objetivo de disminuir el número de posibles vulnerabilidades de seguridad en el mismo.
8.29	Pruebas de seguridad en el desarrollo y la aceptación	Verificar si se satisfacen los requisitos de seguridad de la información al implementar aplicaciones en entornos de producción.
8.31	Separación de los entornos de desarrollo, prueba y producción	Proteger el entorno de producción y los datos frente a posibles amenazas derivadas de las actividades de desarrollo y pruebas.

En resumen, de la norma ISO/IEC 27002:2022 se han seleccionado 26 controles distribuidos de la siguiente forma: 5 controles organizacionales, 1 control de personas, 1 control físico y 18 controles tecnológicos.

5.2. Recomendaciones Aplicables de NIST

La versión 1.1 del Marco de Desarrollo de Software Seguro (SSDF) publicado por NIST propone una serie de recomendaciones para mitigar el riesgo de las vulnerabilidades de software. Estas recomendaciones se encuentran organizadas en 4 grupos: Preparar la organización (PO), Proteger el software (PS), Producir software bien seguro (PW) y Responder a las vulnerabilidades (RV).

De dichas recomendaciones, se priorizan las siguientes:

Tabla 11: Descripción de recomendaciones aplicables de NIST

ID	Recomendación	Descripción
PO1	Definir los requisitos de seguridad para el desarrollo de software	Garantizar que los requisitos de seguridad para el desarrollo de software sean conocidos en todo momento, de modo que puedan tenerse en cuenta a lo largo de todo el ciclo de vida del desarrollo de software, y se pueda minimizar la duplicación de esfuerzos.
PO2	Implementar roles y responsabilidades	Asegurarse de que todas las personas dentro y fuera de la organización que estén involucradas en el ciclo de vida del desarrollo de software estén preparadas para desempeñar sus roles y responsabilidades.
PO5	Implementar y mantener entornos seguros para el desarrollo de software	Asegurarse de que todos los componentes de los entornos de desarrollo de software estén debidamente protegidos contra amenazas internas y externas.
PS1	Proteger todas las formas de código contra el acceso no autorizado y la manipulación	Se destaca la importancia de prevenir modificaciones no autorizadas en el código, tanto si son involuntarias como deliberadas, con el objetivo de salvaguardar las características de seguridad previstas.
PW1	Diseñar software para cumplir con los requisitos de seguridad y mitigar los riesgos de seguridad	Identificar y evaluar los requisitos y riesgos de seguridad del software, así como de diseñar el software de manera que mitigue eficazmente esos riesgos.
PW5	Crear código fuente siguiendo prácticas de codificación segura	Disminuir las vulnerabilidades de seguridad en el software y reducir costos al minimizar la introducción de vulnerabilidades durante la creación del código fuente.
PW8	Probar el código ejecutable para identificar vulnerabilidades y verificar el cumplimiento de los requisitos de seguridad	Identificar y corregir vulnerabilidades antes del lanzamiento del software para prevenir su explotación.

PW9	Configurar el software con ajustes seguros de manera predeterminada	Mejorar la seguridad del software durante la instalación para evitar la posibilidad de que el software sea desplegado con configuraciones de seguridad deficientes.
RV3	Analizar vulnerabilidades para identificar sus causas fundamentales	Analizar las vulnerabilidades identificadas para determinar las causas subyacentes a lo largo del tiempo con el fin de identificar posibles patrones.

5.3. Asignación de Controles a los Riesgos

La siguiente tabla muestra los controles de la norma ISO/IEC 27002:2022 asociados a los riesgos que presenta la DTI en cuanto al proceso de desarrollo y despliegue de aplicaciones.

Tabla 12: Asignación de controles a los riesgos

Código activo	Vulnerabilidad	Control
D01	No documentar las reglas de implementación y autorización durante el proceso de desarrollo	8.31 Separación de los entornos de desarrollo, prueba y producción 8.9 Gestión de la configuración
HW01	Mala asignación de recursos por usuario o servicio	8.6 Gestión de la capacidad
HW01	Falta de mantenimiento preventivo de equipos	7.13 Mantenimiento de equipos
HW01	Falta de uso de certificados SSL para activar protocolo HTTPS	8.20 Seguridad de las redes
HW01	Falta de hardening de seguridad	8.27 Principios de arquitectura y diseño de sistemas seguros 8.9 Gestión de la configuración
HW02	Falta de mantenimiento preventivo de equipos	7.13 Mantenimiento de equipos
P01	Errores de codificación	8.28 Codificación segura

P01	Uso de contraseñas débiles o predeterminadas	5.17 Información de autenticación
P01	Apertura de archivos maliciosos	8.7 Protección contra el malware
P01	Falta de capacitación y conocimientos de programación	6.3 Sensibilización, educación y formación en materia de seguridad de la información
SW01	Uso de componentes o librerías con vulnerabilidades conocidas o sin mantenimiento	8.28 Codificación segura
SW01	No usar patrones de diseño de software	8.26 Requisitos de seguridad de aplicaciones 8.27 Principios de arquitectura y diseño de sistemas seguros
SW01	No usar repositorios para versionamiento del código fuente	8.4 Acceso al código fuente
SW01	No separar las configuraciones de los ambientes de desarrollo y producción	8.9 Gestión de la configuración 8.31 Separación de los ambientes de desarrollo, prueba y producción
SW01	Uso de algoritmos criptográficos obsoletos o débiles	8.24 Uso de criptografía 8.28 Codificación segura
SW01	No usar herramientas de análisis de código estático o dinámico	8.29 Pruebas de seguridad en desarrollo y aceptación
SW02	Manipulación de metadatos como JWT o cookies	8.31 Separación de los entornos de desarrollo, prueba y producción 8.9 Gestión de la configuración
SW02	No contar con ambientes de desarrollo y producción	8.31 Separación de los entornos de desarrollo, prueba y producción 8.9 Gestión de la configuración
SW02	Usar valores por defecto de las configuraciones de seguridad	8.28 Codificación segura
SW02	Exponer APIs inseguros	8.28 Codificación segura
SW02	No realizar pruebas unitarias y de integración	8.29 Pruebas de seguridad en el desarrollo y la aceptación
SW02	Permitir múltiples registros fallidos de autenticación	8.5 Autenticación segura
SW02	No usar captcha en formularios	8.28 Codificación segura 8.5 Autenticación segura

SW02	No expirar las sesiones	8.5 Autenticación segura
SW02	Asignación de privilegios incorrecta	8.2 Derechos de acceso privilegiados
SW02	Elevación de privilegios	8.2 Derechos de acceso privilegiados
SW02	Configuración incorrecta de las cabeceras HTTP	8.9 Gestión de la configuración 8.28 Codificación segura
SW02	No implementar autenticación multifactor	8.5 Autenticación segura
SW02	No validar la entrada de datos	8.26 Requisitos de seguridad de aplicaciones 8.27 Principios de arquitectura y diseño de sistemas seguros
SW02	No utilizar ORM para acceder a la base de datos	8.28 Codificación segura
SW02	No cifrar datos sensibles	8.11 Enmascaramiento de datos
SW02	Presentar mensajes de error detallados	8.28 Codificación segura
SW03	Falta de respaldos de la base de datos	8.13 Información de respaldo
SW03	No contar con ambientes de desarrollo y producción	8.31 Separación de los entornos de desarrollo, prueba y producción 8.9 Gestión de la configuración
SW03	Usar valores por defecto de las configuraciones de seguridad	8.9 Gestión de la configuración
SW03	Asignación de privilegios incorrecta	8.2 Derechos de acceso privilegiados
SW03	Almacenamiento de contraseñas en texto plano	8.11 Enmascaramiento de datos
SW04	Uso de algoritmos criptográficos obsoletos o débiles	8.24 Uso de criptografía
SW05	No generar alertas de actividad inusual	8.15 Registro 8.16 Actividades de monitoreo
SW05	No respaldar periódicamente los logs	8.13 Información de respaldo
SW05	No registrar inicios de sesión exitosos y fallidos	8.15 Registro
SW05	No registrar errores de las aplicaciones	8.15 Registro
SW05	Exponer datos sensibles en los logs	8.15 Registro

CAPÍTULO VI

MARCO DE SEGURIDAD

6.1. Definición de Prioridad de Riesgos y Controles

Para priorizar los riesgos más críticos, en la tabla 13 se detallan las vulnerabilidades asociadas con sus riesgos ordenados de manera descendente según la valoración de cada riesgo en función del impacto y probabilidad.

Tabla 13: Prioridad de los riesgos

Código activo	Vulnerabilidad	P	I	Riesgo	Control
P01	Errores de codificación	3	4	12	8.28
SW02	No implementar autenticación multifactor	4	3	12	8.5
D01	No documentar las reglas de implementación y autorización durante el proceso de desarrollo	3	3	9	8.31 8.9
P01	Uso de contraseñas débiles o predeterminadas	3	3	9	5.17
SW01	No usar herramientas de análisis de código estático o dinámico	3	3	9	8.29

SW05	No generar alertas de actividad inusual	3	3	9	8.15 8.16
HW01	Mala asignación de recursos por usuario o servicio	2	4	8	8.6
HW01	Falta de mantenimiento preventivo de equipos	2	4	8	7.13
SW01	No separar las configuraciones de los ambientes de desarrollo y producción	2	4	8	8.9 8.31
SW02	Usar valores por defecto de las configuraciones de seguridad	2	4	8	8.28
SW02	Exponer APIs inseguros	2	4	8	8.28
SW02	No expirar las sesiones	2	4	8	8.5
SW03	Usar valores por defecto de las configuraciones de seguridad	2	4	8	8.9
SW03	Almacenamiento de contraseñas en texto plano	2	4	8	8.11
SW05	No respaldar periódicamente los logs	2	4	8	8.13
HW02	Falta de mantenimiento preventivo de equipos	2	3	6	7.13
P01	Apertura de archivos maliciosos	2	3	6	8.7
SW01	Uso de algoritmos criptográficos obsoletos o débiles	2	3	6	8.24 8.28
SW02	No realizar pruebas unitarias y de integración	2	3	6	8.29
SW02	Permitir múltiples registros fallidos de autenticación	2	3	6	8.5
SW02	No usar captcha en formularios	2	3	6	8.28 8.5
SW02	No validar la entrada de datos	2	3	6	8.26 8.27
SW05	No registrar inicios de sesión exitosos y fallidos	2	3	6	8.15
SW05	No registrar errores de las aplicaciones	2	3	6	8.15
HW01	Falta de uso de certificados SSL para activar protocolo HTTPS	2	2	4	8.2
HW01	Falta de hardening de seguridad	1	4	4	8.27 8.9
P01	Falta de capacitación y conocimientos de programación	2	2	4	6.3

SW01	Uso de componentes o librerías con vulnerabilidades conocidas o sin mantenimiento	2	2	4	8.28
SW01	No usar repositorios para versionamiento del código fuente	1	4	4	8.4
SW02	Manipulación de metadatos como JWT o cookies	1	4	4	8.31 8.9
SW02	Asignación de privilegios incorrecta	1	4	4	8.2
SW02	Elevación de privilegios	1	4	4	8.2
SW02	Configuración incorrecta de las cabeceras HTTP	2	2	4	8.9 8.28
SW02	Presentar mensajes de error detallados	2	2	4	8.28
SW03	Falta de respaldos de la base de datos	1	4	4	8.13
SW03	No contar con ambientes de desarrollo y producción	1	4	4	8.31 8.9
SW03	Asignación de privilegios incorrecta	1	4	4	8.2
SW05	Exponer datos sensibles en los logs	1	4	4	8.15
SW01	No usar patrones de diseño de software	3	1	3	8.26 8.27
SW02	No contar con ambientes de desarrollo y producción	1	3	3	8.31 8.9
SW02	No utilizar ORM para acceder a la base de datos	1	3	3	8.28
SW02	No cifrar datos sensibles	1	3	3	8.11
SW04	Uso de algoritmos criptográficos obsoletos o débiles	1	3	3	8.24

Criterios de Aceptación de Riesgos

De los 43 riesgos identificados en la tabla 13, los riesgos que poseen un nivel bajo, es decir con una valoración de 1 a 3, son aceptados por la organización; mientras que los riesgos de nivel medio, importante y grave no son aceptados, por lo que deben ser analizados y requieren de un tratamiento adecuado para minimizar su criticidad.

Tabla 14: Criterios de aceptación de riesgos

Nivel	Calificación	Cantidad	Criterio
Bajo	1 a 3	5	Aceptado
Medio	4 a 7	23	No aceptado
Importante	8 a 11	13	No aceptado
Grave	12 a 16	2	No aceptado

6.2. Definición de Roles y Responsabilidades

En el presente trabajo la organización representa a la Dirección de Tecnologías de la Información (DTI) de una Institución de Educación Superior (IES) pública en Ecuador, en el cual se propone que la organización debe estar conformada por 4 departamentos internos de trabajo: Departamento de Desarrollo de Aplicaciones (DDA), Departamento de Seguridad y Redes (DSR), Departamento de Infraestructura (DI), Departamento de Soporte Técnico (DST).

Es necesario que el DDA trabaje en conjunto con los demás departamentos complementarios mencionados para garantizar el éxito del desarrollo, despliegue y continuidad de los proyectos de software. Por tal motivo, se considera que los siguientes roles son fundamentales para este propósito:

Tabla 15: Definición de roles y responsabilidades

Rol	Responsabilidades	Departamento
Director de Tecnologías de la Información	Supervisar el ciclo de vida completo del desarrollo de aplicaciones, desde la concepción hasta la entrega y el mantenimiento. Verificar que se cumplan los alcances establecidos para el desarrollo de aplicaciones.	Todos
Director de Desarrollo de Aplicaciones	Gestionar y definir el alcance de los proyectos de desarrollo de aplicaciones.	DDA
Arquitecto de Soluciones	Diseñar la estructura general de las aplicaciones, incluyendo la selección de tecnologías y patrones de diseño.	DDA
Líder de Desarrollo	Guiar a los desarrolladores con buenas prácticas y uso de herramientas durante el ciclo de vida del desarrollo de software.	DDA
Desarrollador	Escribir líneas de código para crear o modificar aplicaciones web, de escritorio y móviles.	DDA
Administrador de Base de Datos	Supervisar las interacciones con las bases de datos, así como la creación y alteración de registros.	DI
Responsable de Infraestructura	Implementar y gestionar la infraestructura tecnológica que soporta las aplicaciones informáticas y bases de datos de la organización.	DI
Responsable de Seguridad	Revisar que las aplicaciones estén protegidas dentro del ciclo de vida de desarrollo de software.	DSR
Responsable de Soporte Técnico	Verificar que los equipos informáticos de la DTI se encuentren en buenas condiciones.	DST

De los roles mencionados, es importante destacar que se ha dado prioridad a los roles directamente involucrados en el desarrollo y despliegue de aplicaciones. Por ello, se especifican cuatro roles clave para el DDA, mientras que en los otros departamentos se han asignado

uno o dos responsables para abarcar las tareas correspondientes a esas funciones.

Considerando los roles y responsabilidades previamente establecidas, ahora se presenta una tabla que asigna a cada riesgo su responsable correspondiente.

Tabla 16: Asignación de responsables a los riesgos

Código activo	Vulnerabilidad	Riesgo	Control	Responsable
D01	No documentar las reglas de implementación y autorización durante el proceso de desarrollo	9	8.31 8.9	Director de Desarrollo de Aplicaciones
HW01	Mala asignación de recursos por usuario o servicio	8	8.6	Responsable de Infraestructura
HW01	Falta de mantenimiento preventivo de equipos	8	7.13	Responsable de Soporte Técnico
HW01	Falta de uso de certificados SSL para activar protocolo HTTPS	4	8.2	Responsable de Infraestructura
HW01	Falta de hardening de seguridad	4	8.27 8.9	Responsable de Seguridad
HW02	Falta de mantenimiento preventivo de equipos	6	7.13	Responsable de Soporte Técnico
P01	Errores de codificación	12	8.28	Desarrollador
P01	Uso de contraseñas débiles o predeterminadas	9	5.17	Desarrollador, Líder de desarrollo
P01	Apertura de archivos maliciosos	6	8.7	Director de Tecnologías de la Información
P01	Falta de capacitación y conocimientos de programación	4	6.3	Director de Desarrollo de Aplicaciones
SW01	No separar las configuraciones de los	8	8.9 8.31	Desarrollador

	ambientes de desarrollo y producción			
SW01	Uso de componentes o librerías con vulnerabilidades conocidas o sin mantenimiento	4	8.28	Líder de desarrollo, Desarrollador
SW01	No usar repositorios para versionamiento del código fuente	4	8.4	Líder de desarrollo, Desarrollador
SW01	No usar patrones de diseño de software	3	8.26 8.27	Arquitecto de Soluciones, Desarrollador
SW01	No usar herramientas de análisis de código estático o dinámico	9	8.29	Responsable de Seguridad
SW01	Uso de algoritmos criptográficos obsoletos o débiles	6	8.24 8.28	Arquitecto de Soluciones, Desarrollador
SW02	Manipulación de metadatos como JWT o cookies	4	8.31 8.9	Desarrollador
SW02	No contar con ambientes de desarrollo y producción	3	8.31 8.9	Líder de desarrollo, Responsable de Infraestructura
SW02	No implementar autenticación multifactor	12	8.5	Arquitecto de Soluciones, Desarrollador
SW02	Usar valores por defecto de las configuraciones de seguridad	8	8.28	Desarrollador, Líder de desarrollo
SW02	Exponer APIs inseguros	8	8.28	Arquitecto de Soluciones, Desarrollador
SW02	No expirar las sesiones	8	8.5	Líder de desarrollo, Desarrollador
SW02	No realizar pruebas unitarias y de integración	6	8.29	Desarrollador
SW02	Permitir múltiples registros fallidos de autenticación	6	8.5	Arquitecto de Soluciones, Desarrollador
SW02	No usar captcha en formularios	6	8.28 8.5	Arquitecto de Soluciones

SW02	No validar la entrada de datos	6	8.26 8.27	Desarrollador
SW02	Asignación de privilegios incorrecta	4	8.2	Desarrollador, Líder de desarrollo
SW02	Elevación de privilegios	4	8.2	Desarrollador, Líder de desarrollo
SW02	Configuración incorrecta de las cabeceras HTTP	4	8.9 8.28	Responsable de Infraestructura
SW02	Presentar mensajes de error detallados	4	8.28	Desarrollador
SW02	No utilizar ORM para acceder a la base de datos	3	8.28	Arquitecto de Soluciones, Desarrollador
SW02	No cifrar datos sensibles	3	8.11	Líder de desarrollo, Desarrollador
SW03	Falta de respaldos de la base de datos	4	8.13	Administrador de base de datos, Responsable de Infraestructura
SW03	No contar con ambientes de desarrollo y producción	4	8.31 8.9	Administrador de base de datos, Responsable de Infraestructura
SW03	Usar valores por defecto de las configuraciones de seguridad	8	8.9	Administrador de base de datos
SW03	Almacenamiento de contraseñas en texto plano	8	8.11	Administrador de base de datos
SW03	Asignación de privilegios incorrecta	4	8.2	Administrador de base de datos
SW04	Uso de algoritmos criptográficos obsoletos o débiles	3	8.24	Arquitecto de Soluciones, Desarrollador
SW05	No generar alertas de actividad inusual	9	8.15 8.16	Responsable de Seguridad
SW05	No respaldar periódicamente los logs	8	8.13	Responsable de Infraestructura
SW05	No registrar inicios de sesión exitosos y fallidos	6	8.15	Arquitecto de Soluciones, Desarrollador

SW05	No registrar errores de las aplicaciones	6	8.15	Arquitecto de Soluciones, Desarrollador
SW05	Exponer datos sensibles en los logs	4	8.15	Desarrollador

6.3. Elaboración de las Políticas de Seguridad

Con base en el trabajo previamente realizado, en esta sección se procederá a definir las políticas de seguridad a seguir dentro de la Dirección de Tecnologías de la Información (DTI) de una Institución de Educación Superior (IES) pública en Ecuador, para gestionar de manera efectiva el personal que labora en la organización, así como la optimización de los procesos de desarrollo y despliegue de las aplicaciones informáticas.

6.3.1. Política de Gestión del Personal

Objetivo:

La presente política tiene como objetivo definir los lineamientos que los empleados deben cumplir en cuanto a normas internas, entrada y salida de personal, para garantizar un ambiente de trabajo productivo en la organización.

Alcance:

La presente política está dirigida a todos los integrantes de la DTI: Director de Tecnologías de la Información, Director de Desarrollo

de Aplicaciones, Arquitecto de Soluciones, Líder de Desarrollo, Desarrollador, Administrador de Base de Datos, Responsable de Infraestructura, Responsable de Seguridad, Responsable de Soporte Técnico.

Especificaciones:

Normativa Interna

- No está permitido consumir alimentos en los puestos de trabajo de los empleados, ya que se podrían afectar los equipos informáticos por los residuos de comida o bebida. En su lugar, se debe utilizar los espacios asignados para poder comer dentro de las instalaciones de la organización.
- Se prohíbe fumar e ingerir bebidas alcohólicas en cualquier área dentro de las instalaciones de la organización.
- Se debe evitar hablar o conversar en voz alta en los puestos de trabajo dado que esto puede causar distracciones a los demás empleados con sus actividades laborales.
- La laptop de trabajo de cada empleado es exclusivamente para uso laboral dentro de las instalaciones de la organización. En caso de que se necesite utilizar la laptop

fuera de las instalaciones para realizar trabajo remoto, se debe solicitar un permiso al Director de Tecnologías de la Información.

- Los empleados deben asistir a las reuniones de trabajo programadas ya sean en modalidad presencial o virtual. Si por alguna razón el empleado no puede participar en la reunión, se tiene que notificar al jefe inmediato.
- No está permitido que los empleados realicen actividades ajenas a sus responsabilidades asignadas durante su jornada laboral, como trabajar en proyectos personales o externos.
- Los empleados deben asegurarse de bloquear la sesión de su computadora o laptop de trabajo cada vez que se alejen de su puesto, ya que no está permitido dejar la sesión abierta.
- Se prohíbe instalar programas externos en los equipos de trabajo de la organización sin contar con la autorización del jefe inmediato.
- La organización se reserva el derecho a la propiedad intelectual del código desarrollado por sus empleados en las diversas aplicaciones informáticas que están bajo su responsabilidad.

Entrada de Personal

- Se debe explicar al empleado cómo funciona la cultura de la organización.
- Se debe proveer al empleado un espacio dentro de la organización incluyendo las herramientas necesarias para que pueda realizar las actividades que se le han asignado.
- El empleado debe recibir un proceso de inducción sobre las herramientas y tecnologías necesarias para el cargo que ha sido asignado, el cual será impartido por la persona delegada por el jefe inmediato.
- El empleado es responsable de cuidar los equipos informáticos asignados para su puesto de trabajo. Si un equipo empieza a presentar problemas se debe notificar al DST.
- El empleado tiene que firmar un acuerdo de confidencialidad asegurando que no divulgará o no hará mal uso de toda la información sensible vinculada con los activos de la organización.

Salida de Personal

- El empleado responsable de una aplicación informática debe notificar su salida con al menos dos semanas de anticipación. Tras la notificación, el jefe inmediato

asignará a un nuevo responsable, y el responsable saliente deberá capacitar al nuevo responsable.

- Si el responsable de una aplicación informática se retira de la organización sin notificación previa, el jefe inmediato deberá asignar un nuevo responsable, evaluando las capacidades y carga laboral de su equipo a cargo.
- Posterior a la salida del empleado, se debe retirar todos los permisos de acceso concedidos (código fuente, aplicaciones, bases de datos, etc.) de forma inmediata.
- El empleado saliente tiene que devolver todos los equipos informáticos al DST. Después, un responsable de soporte técnico deberá realizar el tratamiento seguro de eliminación de la información y, si es necesario, la eliminación del equipo.

6.3.2. Política de Desarrollo Seguro de Aplicaciones

Objetivo:

La presente política tiene como objetivo garantizar la codificación segura del software, ya sea desarrollado internamente o por terceros, para mitigar riesgos de seguridad a través de la implementación de buenas prácticas de codificación, con la finalidad de mantener la integridad y confidencialidad de la información procesada por las aplicaciones informáticas.

Alcance:

La presente política está dirigida a todo el personal del DDA (Director de Desarrollo de Aplicaciones, Arquitecto de Soluciones, Líder de Desarrollo y Desarrollador), Responsable de Infraestructura y Responsable de Seguridad.

Especificaciones:

Autenticación y Autorización

- Se debe generar contraseñas seguras, ya sea para cuentas institucionales o cuentas individuales, de acuerdo con las siguientes indicaciones:
    - La longitud de la contraseña debe tener mínimo 10 caracteres y máximo 12 caracteres.
    - Tener al menos una letra minúscula, una letra mayúscula y un número.
    - Solo se permite lo siguientes caracteres especiales: @ , * , . , _ , -
    - La contraseña no puede ser similar al nombre de usuario institucional.
    - La contraseña no puede contener los nombres de la persona.

- Las contraseñas que se insertan en la base de datos de las aplicaciones tienen que estar cifradas con un algoritmo de hashing robusto.
- Se prohíbe almacenar contraseñas en texto plano, o contraseñas generadas con algoritmos débiles.
- Se debe almacenar un registro histórico de las contraseñas ingresadas por los usuarios. En caso de renovación, las contraseñas no deben ser idénticas a las 3 últimas registradas.
- Las contraseñas deben tener un tiempo de caducidad para obligar a los usuarios a cambiar su contraseña periódicamente. Se sugiere que sea cada 6 meses.
- Se debe implementar el cierre de sesión automático después de un período de inactividad de 15 minutos del usuario.
- Solo se permiten 5 intentos fallidos para iniciar sesión en las aplicaciones, luego de eso se procede con el bloqueo de la cuenta del usuario y se le envía una notificación al correo personal para que cambie su contraseña.
- Se debe implementar autenticación multifactor en los sistemas críticos, como el envío de un código OTP de verificación al correo electrónico del usuario.

- Todo formulario de registro o contacto de libre acceso debe incorporar un componente de captcha como medida de seguridad para prevención de spam y ataques de fuerza bruta.
- Todo formulario o servicio desarrollado que no sea de libre acceso debe tener permisos de autenticación por Cookie, JWT, Bearer Token, entre otros mecanismos de verificación.
- Se debe proporcionar a los usuarios los privilegios de acceso estrictamente necesarios a las aplicaciones y bases de datos.
- El Director de Desarrollo de Aplicaciones debe supervisar regularmente los privilegios de acceso concedidos a los administradores de las aplicaciones informáticas.

Acceso al código fuente

- Es obligatorio emplear un sistema de control de versiones para alojar el código fuente de las aplicaciones. Por ejemplo: TFS, Gitlab.
- El Director de Desarrollo de Aplicaciones debe gestionar adecuadamente los permisos de lectura y escritura otorgados a los desarrolladores para acceder a los repositorios de las aplicaciones.

- Se debe implementar un proceso de revisión para aprobar los cambios realizados en el código fuente y poder subir dichos cambios al repositorio.
    - El desarrollador tiene que crear una rama para implementar nuevas funcionalidades o correcciones.
    - El desarrollador debe realizar pruebas unitarias y de integración para asegurarse de no cometer errores.
    - El líder de desarrollo debe revisar el código elaborado por el desarrollador, y escanear el código usando una herramienta de análisis estático.
    - Si el líder de desarrollo encuentra algún error o vulnerabilidad, el desarrollador a cargo deberá implementar las modificaciones necesarias.
    - Si ya no se presentan novedades, el código será fusionado con la rama principal del repositorio.

Codificación y Ciclo de Vida del Software

Antes de la Codificación

- El Director de Desarrollo de Aplicaciones debe elaborar guías de codificación segura para cada lenguaje de

programación utilizado en las diferentes aplicaciones informáticas.

- El Director de Desarrollo de Aplicaciones debe capacitar regularmente a los desarrolladores en aspectos de codificación segura considerando las últimas amenazas y técnicas de intrusión.
- El arquitecto de soluciones debe seleccionar las tecnologías y frameworks adecuados en función de las características y requisitos de la aplicación que se va a desarrollar.
- El líder de desarrollo debe realizar un análisis del alcance y el impacto que tendrá el desarrollo de un nuevo requerimiento para no afectar a otras aplicaciones.
- El desarrollador debe instalar las herramientas necesarias para la programación de código, como el uso de un Entorno de Desarrollo Integrado (IDE).
- El desarrollador debe asegurarse que las librerías o componentes que se pretenden utilizar dentro de una aplicación no incluyan vulnerabilidades que puedan provocar un riesgo. En caso de que su uso sea estrictamente necesario se debe notificar al Arquitecto de Soluciones y al Director de Desarrollo de Aplicaciones.

Durante la Codificación

- El desarrollador debe programar empleando técnicas de codificación segura, para prevenir ataques de intrusión comunes. Esto incluye:
    - Validar los datos introducidos en los formularios tanto en el front-end como en el back-end.
    - Utilizar una herramienta de Mapeo Objeto-Relacional (ORM) para interactuar con la base de datos.
    - Aplicar patrones de diseño acorde a la necesidad de cada modelo de negocio, por ejemplo, el uso del patrón “Singleton” para garantizar una sola instancia de conexión a la base de datos.
    - Configurar adecuadamente las cabeceras HTTP, como el uso de “Access-Control-Allow-Origin" para controlar el acceso a los recursos de la aplicación.
- En aplicaciones robustas, el back-end y el front-end de la aplicación tiene que estar desacoplado en proyectos independientes.
- Se debe gestionar las sesiones de manera eficiente incorporando el uso de tokens seguros.

- El desarrollador debe implementar el manejo de excepciones para gestionar errores y garantizar el funcionamiento continuo de la aplicación, evitando problemas de disponibilidad.
- El desarrollador no debe exponer información sensible (datos de usuario, errores de sistema, versión de framework, etc.) en mensajes de información, advertencia o error.
- El desarrollador debe registrar los eventos importantes en un archivo de log para mantener un historial detallado.
- Se debe almacenar la configuración de la aplicación en un archivo global que esté separado del código principal.
- Es obligatorio que el desarrollador evalúe la seguridad del código implementado utilizando herramientas de análisis estático de código.

Después de la Codificación

- El desarrollador tiene que elaborar un manual de usuario con el detalle de las funcionalidades de la aplicación.
- El desarrollador debe documentar un manual técnico que incluya las reglas de implementación y autorización empleadas en la aplicación como:
    - Registro de cambios realizados.

  - Nombres y versiones de las dependencias (librerías, frameworks, etc.) utilizadas.
  - Detalle de permisos hacia otras aplicaciones o bases de datos.
- El Director de Desarrollo de Aplicaciones debe mantener un inventario de software actualizado de todas las aplicaciones implementadas, que incluya información relevante como: nombre de la aplicación, desarrolladores o responsables de la aplicación, lenguajes de programación y componentes utilizados, entre otras especificaciones técnicas.
- El responsable de seguridad debe realizar pruebas de penetración y un análisis de vulnerabilidades de la aplicación, empleando herramientas especializadas.
- El desarrollador debe actualizar regularmente los componentes de terceros usados en la aplicación, como librerías, bibliotecas y frameworks. De ser necesario, se debe planificar una ventana de mantenimiento para llevar a cabo las actualizaciones.

Registros de Eventos (Logs)

- La hora del sistema de los servidores tiene que estar sincronizada en todas las aplicaciones con la zona horaria regional que es UTC-5.
- Se debe registrar todo tipo de evento que se considere importante según la criticidad de la aplicación o sistema. Por ejemplo:
    - Accesos a las aplicaciones.
    - Intentos de inicio de sesión fallidos y exitosos.
    - Errores en tiempo de ejecución.
- El desarrollador tiene que evitar almacenar información sensible en los registros que pueda ser explotada.
- El responsable de infraestructura debe implementar medidas de protección contra manipulaciones no autorizadas como la eliminación o alteración de registros.
- El responsable de seguridad tiene que monitorear los registros en busca de actividades inusuales o amenazas, con la ayuda de herramientas de escaneo.
- El responsable de infraestructura tiene que realizar copias de seguridad semanales de los registros y configurar que el periodo de retención de los registros sea por lo menos de 6 meses.

- El responsable de infraestructura debe eliminar los registros antiguos de manera segura tomando en cuenta la capacidad de almacenamiento de los servidores.

6.3.3. Política de Despliegue de Aplicaciones

Objetivo:

Esta política tiene como objetivo garantizar la seguridad y eficiencia del proceso de despliegue o pase a producción de las aplicaciones informáticas para asegurar su disponibilidad. Además, se busca optimizar recursos y minimizar riesgos por ataques de intrusión con la finalidad de que las aplicaciones funcionen correctamente y se mantengan actualizadas.

Alcance:

Esta política está dirigida al personal del DDA (Director de Desarrollo de Aplicaciones, Arquitecto de Soluciones, Líder de Desarrollo y Desarrollador), Administrador de Base de Datos, Responsable de Infraestructura y Responsable de Seguridad.

Especificaciones:

Gestión de Ambientes

- La implementación de toda aplicación se debe manejar dentro de 3 ambientes de trabajo: desarrollo, pruebas y producción.

- Todos los desarrolladores tienen que programar el código de las aplicaciones específicamente en el ambiente de desarrollo.
- Se tiene que separar adecuadamente los ambientes de trabajo, ya que no se permite configurar aplicaciones que aún se encuentran en fase de desarrollo en un ambiente de producción.
- Se tiene que verificar el correcto funcionamiento de una aplicación dentro del ambiente de pruebas, y asegurarse de que se repliquen las mismas configuraciones hacia el ambiente de producción para prevenir fallos de funcionalidades.
- Los dominios internos o de prueba que se usan en las aplicaciones tienen que ser creados en un ambiente de prueba, con la asignación de una dirección ip privada. No se permite crear un dominio de prueba en un ambiente de producción.
- No está permitido subir un cambio directamente al ambiente de producción sin haber evaluado primero la funcionalidad en un ambiente de pruebas, a menos que la circunstancia lo justifique y se cuente con la aprobación del Director de Desarrollo de Aplicaciones.

- Los desarrolladores o responsables de una aplicación informática deben encargarse personalmente de llevar a cabo el despliegue de la aplicación. Si el despliegue debe realizarse con urgencia y el responsable no está disponible, su jefe inmediato asumirá esta gestión.
- Solamente el personal autorizado debe tener acceso a los ambientes de producción de las diferentes aplicaciones.
- Los desarrolladores no deben tener acceso a las bases de datos del ambiente de producción, ya que ese privilegio es exclusivo para los administradores de base datos.
- En su lugar, los desarrolladores podrán acceder a una base de datos de prueba que sea una réplica del ambiente productivo configurada por el administrador de base de datos.

Antes del despliegue a producción

- Se debe documentar los cambios realizados durante el desarrollo de las aplicaciones para mantener un historial detallado.
- Es obligatorio realizar un proceso de hardening en las aplicaciones y en los servidores que funcionarán para el ambiente de producción. Esto incluye eliminar servicios

innecesarios y configurar las opciones de seguridad requeridas.

- Se debe eliminar todas las funciones de mensajes informativos (prints) en la aplicación para que no sean visibles posterior al despliegue.
- Se debe ejecutar un análisis de vulnerabilidades de las aplicaciones dentro del ambiente de pruebas para reducir el riesgo de ataques de intrusión.
- Es obligatorio configurar un certificado ssl para cualquier aplicación que se libera en ambiente de producción, con la finalidad de habilitar el protocolo https y cifrar la información.
- En caso de que no se pueda adquirir un certificado ssl, se debe configurar temporalmente un certificado ssl gratuito para la aplicación, tomando en cuenta que este tipo de certificado tiene que ser renovado cada 3 meses.

Después del despliegue a producción

- Se debe evaluar constantemente el rendimiento de las aplicaciones en ambiente de producción. De ser necesario se debe ajustar los parámetros de configuración de las aplicaciones o bases de datos, así como incrementar los recursos asignados a los servidores.

- Se debe actualizar regularmente las versiones de las dependencias, frameworks y componentes de terceros en las aplicaciones y servidores, asegurándose de tomar las medidas de precaución adecuadas como la ejecución de respaldos o snapshots.
- Se debe ejecutar periódicamente un análisis de vulnerabilidades de las aplicaciones que están en ambiente de producción para identificar nuevas debilidades o problemas de seguridad.
- Se debe configurar alertas para identificar posibles fallos o errores que presentan las aplicaciones.
- Se debe realizar periódicamente mantenimiento preventivo de los servidores que alojan las aplicaciones y base de datos.
- Es obligatorio realizar copias de seguridad programadas de los activos de información:
    - La base de datos se debe respaldar todos los días, de preferencia en horarios de poco tráfico, como en la madrugada.
    - El código fuente tiene que estar respaldado en el sistema de control de versiones. En caso de que la

aplicación no se maneje con versionamiento, se tiene que respaldar el código todos los días.

  - Los logs se deben respaldar una vez por semana.

- Cuando se presente un incidente en una aplicación en producción, se tiene que replicar el error en un ambiente de pruebas empleando los respaldos más recientes de la aplicación y base de datos. Luego de resolver el incidente de manera segura, se debe subir los cambios aplicados a producción.

CONCLUSIONES

1. Al momento de elaborar el presente trabajo de titulación, se pudo constatar que la Dirección de Tecnologías de la Información (DTI) no cuenta con una política de seguridad de la información formalmente establecida, sin embargo, la DTI dispone de documentación sobre procesos y procedimientos, aunque estos no se aplican de manera uniforme ni completa, relacionados con el desarrollo de aplicaciones, administración de bases de datos, gestión de activos, amenazas, vulnerabilidades y riesgos.
2. La identificación de los activos de información y su respectivo análisis de riesgos descrito en el Capítulo 4, destaca la importancia de proteger los recursos de la DTI. Al categorizar las amenazas y vulnerabilidades, y asignarles un nivel de riesgo, se establece la base para priorizar las medidas de seguridad necesarias, asegurando así la integridad, confidencialidad y disponibilidad de las aplicaciones informáticas de la organización.
3. La implementación de controles basados en la norma ISO/IEC 27002:2022 y las recomendaciones de NIST en la Dirección de Tecnologías de la Información (DTI) fortalece significativamente la seguridad del desarrollo y despliegue de aplicaciones. Al identificar y aplicar controles específicos para cada riesgo, la organización no solo mejora su postura de seguridad, sino que también establece un marco

sólido que garantiza que las aplicaciones se encuentren operacionales respetando la confidencialidad, integridad y disponibilidad de la información.

4. En nuestra propuesta de marco de seguridad se definieron 2 políticas claves orientadas a mejorar los procesos de desarrollo y despliegue de las aplicaciones informáticas gestionadas por la DTI: La Política de Desarrollo Seguro de Aplicaciones, que busca garantizar la creación de software seguro y robusto desde sus primeras fases, y la Política de Despliegue de Aplicaciones, enfocada en asegurar un proceso controlado y eficiente para la puesta en producción de las mismas. Además, se elaboró una Política de Gestión del Personal que comprende las normas internas de la DTI y permite a los empleados conocer el entorno de la organización desde el inicio de su trabajo para que puedan desempeñar sus funciones de manera eficaz.

RECOMENDACIONES

1. Se recomienda fortalecer los programas de capacitación y realizar sesiones de talleres para concientizar a todos los miembros de la organización sobre buenas prácticas y temas de seguridad que afecten la disponibilidad, integridad y confidencialidad de las aplicaciones informáticas.
2. Se recomienda llevar a cabo un análisis estático del código fuente de las aplicaciones para evaluar la calidad del código implementado y evitar la introducción de errores o funcionalidades no deseadas.
3. Se recomienda realizar un análisis de vulnerabilidades y pruebas de penetración para identificar y corregir problemas de seguridad en las aplicaciones antes de ser liberadas a producción.
4. Se recomienda implementar un mecanismo de monitoreo en tiempo real que supervise el rendimiento de las aplicaciones y bases de datos, el cual emita alertas en caso de que no se cumplan los requisitos de seguridad.
5. Se recomienda implementar un proceso de análisis y tratamiento de logs, empleando herramientas especializadas, para detectar actividad inusual o sospechosa en las aplicaciones informáticas.
6. Se recomienda automatizar el proceso de despliegue de las aplicaciones utilizando herramientas de Integración Continua y Despliegue Continuo (CI/CD). Esto implica configurar pipelines que

integren pruebas y análisis de brechas de seguridad para validar la aplicación antes de subir cambios al ambiente de producción.

7. Se recomienda mejorar el control de los derechos de acceso privilegiados, ya que se pudo identificar que los desarrolladores interactúan directamente con la base de datos del ambiente de producción, un privilegio que debería estar reservado únicamente para los administradores de bases de datos.
8. Se recomienda incorporar el uso de un sistema de control de versiones en los sitios webs institucionales basados en gestores de contenido como Drupal o Wordpress.
9. Se recomienda que los integrantes del Departamento de Desarrollo de Aplicaciones trabajen de manera coordinada con los responsables del Departamento de Seguridad y Departamento de Infraestructura para garantizar la implementación de aplicaciones seguras y minimizar las brechas de seguridad.
10. Se recomienda elaborar una política de gestión de incidentes que determine las funciones de los responsables y los pasos a seguir para recuperarse de manera efectiva ante eventos inesperados.
11. Se recomienda supervisar el contenido de las políticas elaboradas en la presente tesis cada 6 meses, para ajustarse a las nuevas necesidades de la organización en caso de existir cambios.

BIBLIOGRAFÍA

[1] A. K. Pillay and N. A. Sharma, “Applicable Cyber Security Recommendations to Prevent Cyber Attacks in Universities,” in *2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, 2022, pp. 1–5. doi: 10.1109/CSDE56538.2022.10089360.

[2] International Organization for Standardization (ISO), “ISO/IEC 27002:2022. Information technology — Security techniques — Code of practice for information security controls.” International Organization for Standardization (ISO), 2017.

[3] M. Souppaya, K. Scarfone, and D. Dodson, “Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities.” National Institute of Standards and Technology Special Publication, Feb. 2022. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

[4] International Organization for Standardization (ISO), “ISO/IEC 27000:2016. Information technology – Security techniques – Information security management systems.” International Organization for Standardization (ISO), 2016. [Online]. Available: https://www.iso.org/standard/66435.html

[5] S. Srithar, E. Vetrimani, V. Vignesh, M. S. Ulaganathan, B. R. Kumar, and S. Alagumuthukrishnan, “Cost-Effective Integration and Deployment of Enterprise Application Using Azure Cloud Devops,” in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2022, pp. 01–05. doi: 10.1109/ICCCI54379.2022.9740874.

[6] Microsoft, “Qué es seguridad de la información (InfoSec),” Qué es seguridad de la información (InfoSec). Accessed: Oct. 13, 2023. [Online]. Available: https://www.microsoft.com/es-ww/security/business/security-101/what-is-information-security-infosec

[7] OWASP, “OWASP. OWASP Top 10: 2021.,” OWASP TOP 10. Accessed: Oct. 13, 2023. [Online]. Available: https://owasp.org/Top10/es/

[8] Amazon, “What is SDLC (Software Development Lifecycle),” What is SDLC (Software Development Lifecycle). Accessed: Oct. 13, 2023. [Online]. Available: https://aws.amazon.com/what-is/sdlc

[9] Instituto Nacional de Ciberseguridad (INCIBE), anteriormente Centro Criptológico Nacional (CCN), *Magerit: Metodología de Análisis y Gestión de Riesgos de Sistemas de Información*, vol. 1. CCN, 2012.

[10] International Organization for Standardization (ISO), “ISO 31000:2018 - Risk management — Guidelines.” International Organization for Standardization (ISO), 2018.

[11] International Organization for Standardization (ISO), “ISO/IEC 27001:2017 - Information technology — Security techniques — Information security management systems — Requirements.” International Organization for Standardization (ISO), 2017.

[12] L. ALMAGRO, “MARCO NIST CIBERSEGURIDAD Un abordaje integral de la Ciberseguridad,” *Internet Httpswww Oas OrgessmscictedocsOEA-AWS-Marco-NIST--Ciberseguridad-ESP Pdf*.

[13] G. King and Y. Yingxu, *Software engineering processes: Principles and applications*. CRC Press, 2000.