Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado

UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS ECONÓMICAS
ESCUELA DE ESTUDIOS DE POSGRADO

MAESTRÍA EN CIBERDEFENSA Y CIBERSEGURIDAD

TRABAJO FINAL DE MAESTRÍA

RECOMENDACIONES DE CIBERDEFENSA PARA LA GESTIÓN SEGURA DEL CICLO DE VIDA DE SISTEMAS CRÍTICOS

AUTOR: INGENIERO FEDERICO FERNANDO CÁCERES

DIRECTOR: INGENIERO ANÍBAL LUIS INTINI

[JULIO 2021]

Dedicatorias

A mi familia y amigos que me acompañaron y apoyaron durante estos años.

A Vico por la compañía y el respaldo durante todo este gran proceso.

Agradecimientos

A mis compañeros de la cohorte 2018 con quien compartimos tantas horas, mensajes y reuniones, a mi equipo de cursada María Laura, Victoria, Sofía y Miguel por su apoyo y amistad. Especialmente a Agostina, mi gran amiga, consejera y colega de catedra.

A mi director de Tesis Ing. Aníbal Intini por el tiempo, la dedicación y las enseñanzas brindadas durante el armado de la presente Tesis.

Al cuerpo docente de la Maestría por la predisposición y el profesionalismo.

Resumen

Esta tesis de maestría está incluida en el campo del conocimiento de la Ciberseguridad y Ciberdefensa, específicamente en el subconjunto del desarrollo de aplicaciones y sistemas. Este último contexto, se caracteriza por incluir problemas que pueden mutar en oportunidades de desarrollo de proyectos de investigación que constituyan aportes relevantes. En el ciclo de vida de sistemas críticos, se han detectado problemas que devienen en vulnerabilidades potencialmente explotables. Esto se produce, esencialmente, como resultado de una planificación y ejecución de dicha gestión sin una estructura basada en la incorporación de medidas de seguridad tal como se describe en esta tesis de maestría.

Para sustentar la *propuesta de solución,* se presenta un análisis del estado del arte de las etapas y técnicas actualmente utilizadas, difiriendo la eficacia de éstas con diversos estudios y estadísticas internacionales.

El aporte de la presente tesis a la comunidad técnica, científica y académica es una recomendación a la gestión segura del ciclo de vida de sistemas con principal foco en las Infraestructuras y Sistemas críticos que se encuentran cada vez más expuestos en un mundo hiperconectado.

Abstract

This master's thesis is included in the field of knowledge of Cybersecurity and Cyber Defense, specifically in the subset of application and systems development. The latter context is characterized by the inclusion of problems that may change into opportunities for the development of research projects that constitute relevant contributions. In the life cycle of critical systems, problems have been identified that become potentially exploitable vulnerabilities. This occurs essentially as a result of the planning and implementation of such management without a structure based on the incorporation of security measures as described in this master's thesis.

To support the proposed solution, an analysis of the state of the art of the stages and techniques currently used is presented, differing their effectiveness with various international studies and statistics.

The contribution of this thesis to the technical, scientific and academic community is a recommendation for the safe management of the life cycle of systems with the main focus on Critical Infrastructures and Systems that are increasingly exposed in a hyper-connected world.

Índice General

Índice de Tablas

Índice de Figuras

Abreviaturas

A

AEPD: Agencia Española de Protección de Datos

AES: Estándar de cifrado avanzado.

AGA: Asociación de gas americano (EE. UU.)

API: Interfaces de programación de aplicaciones

APT: Amenazas Avanzadas Persistentes

B

BSIMM-V: Construyendo modelo de seguridad en madurez.

C

CbC: Corrección por construcción.

CC: Criterio Común.

CISSP: Certificación internacional de profesional en seguridad de sistemas de información.

CMM: Modelo de Madurez de Capacidades.

CMMI: Integración de Sistemas de Modelos de Madurez de Capacidades.

CMMI - DEV: Integración de Sistemas de modelos de Madurez de Capacidades para el Desarrollo.

CPU: Unidad Central de Procesamiento.

CTED: Dirección Ejecutiva del Comité contra el Terrorismo de las Naciones Unidas.

D

DDoS: Ataque de Denegación de Servicio Distribuida.

DevSecOps: Paradigma de desarrollo y operaciones con una perspectiva de seguridad.

DFD: Diagrama de Flujo de Datos.

E

EAL: Niveles de garantía de evaluación.

G

GAO: Oficina de Responsabilidad del Gobierno de Estados Unidos.

GDRP: Reglamento General de Protección de Datos de la Unión Europea.

I

IC: Infraestructura Crítica.

IEC: Comisión Electrotécnica Internacional.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

IETF: Grupo de Trabajo de Ingeniería de Internet.

ISA: Sociedad Internacional de Automatización.

ISACA: Asociación de auditoría y control de sistemas de información.

ISO: Organización Internacional de Normalización.

ITIL: Biblioteca de Infraestructuras de Tecnologías de Información.

F

FIPS: Estándar Federal de procesamiento de información (EE. UU.).

N

NCIIPC: Centro Nacional de Protección de Infraestructura de Información Crítica (India).

NCSC: Centro Nacional de Seguridad Cibernética.

NIPP: Plan de Protección de Infraestructuras Nacionales de Estados Unidos.

NIST: Instituto Nacional de Estándares y Tecnología (EE. UU.).

NISTIR: Reporte Interno del Instituto Nacional de Estándares y Tecnología (EE. UU.).

O

OCDE: Organización para la Cooperación y el Desarrollo Económico.

OWASP: Proyecto de seguridad de aplicaciones web abiertas

P

PbD: Privacidad por Defecto.

PET: Tecnologías de mejora de la privacidad.

PII: Identificador de Información Personal.

PITAC: Comité Asesor de Tecnología de la Información del presidente de los EE. UU.

R

RSA: Acrónimo de los creadores de un sistema criptográfico de clave pública: Rivest, Shamir y Adleman.

S

SAFECode: Foro de garantía de software para la excelencia en el código.

SAMM: Modelo de madurez para el aseguramiento del software.

SEI: Instituto de Ingeniería de Software.

SCADA: Supervisión, Control y Adquisición de Datos.

SDLC: Ciclo de vida de desarrollo de software.

SGC: Sistema de gestión de la calidad

SLC: Líneas de código fuente.

SSDL: Ciclo de vida del desarrollo seguro de software.

STRIDE: Modelo de amenazas que contempla Manipulación, Repudio, Revelación de información, Denegación de Servicio.

T

TIC: Tecnologías de la información y las comunicaciones.

TSP: Proceso de software del equipo.

U

UNOCT: Oficina de lucha contra el terrorismo de las Naciones Unidas.

US. DHS: Departamento de Seguridad Nacional de los Estados Unidos.

1. Introducción

El ***ciberespacio*** es el dominio global y dinámico compuesto por las infraestructuras tecnológicas -incluida Internet-, las redes y los sistemas de información y de telecomunicaciones (Decreto 577/2017). Este dominio desafía las capacidades convencionales de los Estados y el costo relativamente bajo de las operaciones cibernéticas - en comparación con las operaciones militares tradicionales - representa una ventana de oportunidad para los agresores (Rodríguez, 2017), esto se ve reflejado en el contexto internacional en donde se observa una proliferación de ***ciberataques***[1] cada vez más sofisticados y potencialmente devastadores (Azzollini, 2017).

La evolución de la arquitectura de las computadoras y sus componentes, como así también la de los estándares y normativas de calidad asociadas, requieren de un particular estudio que permita proponer una mejora en las medidas de seguridad de la Nación. Así lo hace notar la Directiva de Política de Defensa Nacional cuando menciona que "tanto los Estados como los actores no estatales están desarrollando medios cibernéticos para explotar las vulnerabilidades inherentes a los sistemas de comando, control, comunicaciones, inteligencia, vigilancia y reconocimiento" (Decreto 703/2018 - *derogado*[2]).

En este sentido la Unión Internacional de Telecomunicaciones define que:

> "La ciberseguridad es el conjunto de herramientas, políticas, conceptos de seguridad, salvaguardas de seguridad, directrices, métodos de gestión de riesgos, acciones, formación, prácticas idóneas, seguros y tecnologías que pueden utilizarse para proteger los activos de la organización y los usuarios en el ciberentorno." (ITU, 2008)

El Instituto Nacional de Estándares y Tecnología de los Estados Unidos define la *defensa del ciberespacio* como las "acciones normalmente creadas dentro del ciberespacio del Departamento de Defensa para proteger, operar y defender las redes de información del

[1] Se entiende por ciberataque al ataque dirigido al uso del ciberespacio de una empresa con el fin de interrumpir, deshabilitar, destruir o controlar maliciosamente un entorno / infraestructura informática o destruir la integridad de los datos o robar información controlada (CSRC, 2019).

[2] Al derogar este Decreto, y reestablecer la vigencia de los anteriores, se ha perdido la inclusión de los conflictos del ciberespacio. Resulta necesario se actualice la Política de Defensa Nacional en línea con lo que establecía el Decreto 703/2018.

Departamento de Defensa. Las acciones específicas incluyen proteger, detectar, caracterizar, contrarrestar y mitigar" (CSRC, 2019).

En noviembre del 2013 el Consejo Argentino para las Relaciones Internacionales definió la Ciberdefensa como el "conjunto de acciones de defensa activas pasivas, proactivas, preventivas y reactivas para asegurar el uso propio del ciberespacio y negarlo al enemigo o a otras inteligencias en oposición." *(CARI, 2013)*

En el presente trabajo final de maestría se sostiene que la ciberseguridad debe ser definida como el contexto de trabajo de medidas asociadas a la protección del ciberespacio para los usuarios, las organizaciones y el Estado. Para la protección específica de este último y, en mayor medida, de las infraestructuras nacionales es que se debería trabajar bajo la visión de la Ciberdefensa. En contraposición, si se identificara que un actor se encuentra cometiendo un acto ilícito que pudiera afectar dichas infraestructuras, sería apropiado utilizar el término Cibercrimen y la jurisdicción serían las fuerzas de seguridad y la justicia federal dado el contexto argentino.

Recientemente, el 8 de abril del 2021, se creo el Comité Asesor para el Desarrollo e Implementación de Aplicaciones Seguras el cual tiene por objetivo brindar en la elaboración de guías y protocolos de principios y buenas prácticas relacionadas con la seguridad en el desarrollo, contratación e implementación de aplicaciones informáticas utilizadas por los organismos del Sector Público Nacional. (Disposición 6/2021).

El *problema* que existe en la actualidad radica en la multiplicidad de organismos que emiten recomendaciones, normas y buenas prácticas relacionadas con la programación de sistemas y elementos informáticos. Además, los Estados también desarrollan sus propias prácticas, leyes y recomendaciones lo que genera una mayor *dificultad* a la hora de iniciar un proyecto de desarrollo de aplicaciones y sistemas críticos ya que al tener que optar por las distintas recomendaciones, se pone en riesgo la seguridad de estos teniendo en cuenta la importancia que esto tiene al tratarse de sistemas críticos.

En efecto, la falta de estándares que integren todos los puntos antes mencionados con el ciclo de vida del software, las técnicas de aseguramiento de la triada (confidencialidad, integridad y disponibilidad) y los atributos propios de los sistemas que soportan las infraestructuras críticas nacionales permite que se escoja una recomendación y omitiendo el resto sin ponderar los riesgos que pudieran impactar en la seguridad del proyecto y, por ende, en el Estado y las infraestructuras críticas.

El proceso de digitalización y modernización del Estado, junto con el aumento de los ciberataques, permite el incremento de los riesgos de una gran cantidad de sistemas. Estos deben ser desarrollados de forma tal que garanticen solidez y cumplan con todas las características del desarrollo seguro.

El *propósito* del presente trabajo es introducir una recomendación para la gestión segura de todo el ciclo de vida del desarrollo de sistemas críticos que contemple el concepto de privacidad por defecto, el método de corrección por construcción y los conceptos de seguridad por diseño y por defecto, así como también las recomendaciones, normas, estándares y buenas prácticas utilizados en la actualidad. Su aplicabilidad permitiría disminuir las potenciales vulnerabilidades y fallas en tiempos de ejecución, así como también reducir los costos y tiempos de desarrollo de los proyectos de software al detectar problemas con el código.

Para lograrlo se han llevado adelante las siguientes tareas:

- Recopilar distintas normas, recomendaciones, prácticas y metodologías asociadas al desarrollo de software seguro y a la seguridad de las infraestructuras críticas de un país.
- Analizar distintas versiones de ciclos de vida de desarrollo seguro e identificar los puntos más relevantes.

Este trabajo final de maestría esta divido en los siguientes capítulos:

2. Estado del Arte

Este capítulo describe el estado actual del arte, explicando los conceptos relacionados tanto al desarrollo de sistemas y aplicaciones, como a las infraestructuras críticas. Para completar el entendimiento de la situación actual, se realiza un abordaje a las diversas Normas, Estándares y Recomendaciones. Para mayor claridad, se encuentra subdividido en dos secciones:

1. Se plantean los desafíos de ciberdefensa y ciberseguridad en el desarrollo introduciendo los conceptos de Infraestructuras Críticas, Sistemas Críticos y de Desarrollo Seguro.

2. Se presenta un análisis de los antecedentes de Normas, Estándares y Recomendaciones para el Desarrollo de aplicaciones en el marco de la ciberdefensa y ciberseguridad.

3. Aspectos específicos para tener en cuenta en el marco de ciberdefensa para el desarrollo seguro

Se abordan conceptos y recomendaciones especificas a considerar a la hora de desarrollar los códigos utilizados en los Sistemas Críticos.

4. Solución, Implementación y Modelo

Teniendo en consideración lo presentado en la tesis, se exhiben recomendaciones de ciberdefensa para la gestión segura del ciclo de vida de sistemas críticos contemplando los conceptos, métodos, recomendaciones, normas, estándares y buenas prácticas de forma armonizada.

5. Conclusiones y Trabajos Futuros

Se presentan las conclusiones del trabajo y se describen las posibles extensiones de los temas desarrollados, con el objetivo de motivar la investigación de estos tópicos y, en especial, de integrar los analizado en este trabajo con una plataforma de desarrollo para el Estado Nacional, que permita incrementar la seguridad y aplicar los conceptos aquí planteados.

2. Estado del Arte

2.1 Desafíos de ciberdefensa y ciberseguridad en el desarrollo

2.1.1 Infraestructuras críticas (IC).

En el año 2019, se publicó en el boletín oficial de la República Argentina la Resolución 1523 que define que a las infraestructuras críticas como:

> "Aquellas que resultan indispensables para el adecuado funcionamiento de los servicios esenciales de la sociedad, la salud, la seguridad, la defensa, el bienestar social, la economía y el funcionamiento efectivo del Estado, cuya destrucción o perturbación, total o parcial, los afecte y/o impacte significativamente." (Anexo I, p.1)

Asimismo, define que "las Infraestructuras Críticas de Información son las tecnologías de información, operación y comunicación, así como la información asociada, que resultan vitales para el funcionamiento o la seguridad de las Infraestructuras Críticas*"*. (Resolución 1523/2019)

Los sectores identificados son:

- Energía
- Transportes
- Salud
- Finanzas
- Químico
- Espacio
- Estado
- Hídrico
- Alimentación
- Nuclear
- Tecnologías de Información y Comunicaciones

Definición de sectores que comprenden las IC (Resolución 1523/2019, Anexo I, p.2).

Las IC de la información se han vuelto especialmente vulnerables a piratas informáticos, delincuentes e incluso actores estatales y terroristas. Las principales herramientas utilizadas para atacar los sistemas críticos son programas malignos que modifican y destruyen información o bloquean los sistemas informáticos (Nickolov, E. 2005).

El Departamento de Seguridad Nacional de los Estados Unidos define las IC como "los activos, sistemas y redes, ya sean físicos o virtuales, tan vitales que su incapacitación o destrucción tendría un efecto debilitador en la seguridad, la economía nacional, la salud pública o cualquier combinación de ellas" (US DHS & NIST, 2011).

Otras definiciones de IC (CTED & UNOCT, 2018):

- Pakistán: Incluye las infraestructuras así designadas por el gobierno y otros activos, sistemas y redes, ya sean físicos o virtuales, tan vitales para el estado o sus organismos, incluida la judicatura, que su incapacidad o destrucción puede tener un efecto debilitante en la seguridad nacional, economía, salud pública, seguridad operacional o asuntos relacionados.
- Reino de Arabia Saudita (KSA): El sistema y los activos, ya sean físicos o virtuales, tan vitales para el KSA que la incapacidad o destrucción de dichos sistemas y activos tendría un impacto debilitante en la seguridad, la seguridad económica nacional, la salud o seguridad pública nacional, o cualquier combinación de esos asuntos.
- Federación Rusa: Un objeto de infraestructura de importancia crítica es un objeto cuya interrupción (o finalización) de operación conduce a la pérdida de control, destrucción de infraestructura, cambio negativo irreversible (o destrucción) de la economía de un país, un sujeto de la Federación de Rusia, o una unidad administrativa-territorial o un deterioro significativo en la seguridad de la vida de la población que vive en estos territorios durante mucho tiempo.

A diferencia de la Republica Argentina, el Departamento de Seguridad Nacional de los Estados Unidos (US. DHS) propone los siguientes sectores como IC:

1. Químico
2. Instalaciones Comerciales
3. Comunicaciones
4. Crítico de fabricación
5. Represas
6. Base industrial de defensa
7. Servicios de emergencia
8. Energético
9. Servicios financieros
10. Alimentación y Agricultura
11. Instalaciones del Gobierno
12. Sector de Salud y Salud Pública
13. Tecnología de la información
14. Reactores nucleares, materiales y residuos
15. Sistemas de transporte
16. Sistemas de Agua y Aguas Residuales

Definición de sectores que comprenden las IC (US DHS & NIST, 2015, p.10).

En el Reino Unido, esta clasificación se divide en IC e IC Nacionales donde se pondera la criticidad de una por sobre las otras dependiendo sus interrelaciones. Por ejemplo, el sector de la energía tiene un valor inigualable dado que la interrupción de dicho servicio podría afectar las funciones de otros sectores de IC como los servicios de emergencia, las

comunicaciones, la salud y el transporte ocasionando, a su vez, una amenaza para la economía nacional, el orden social y político (Sieńko, P., 2015).

Mosadeghi, Richards y Tomlinson (Mosadeghi et al, 2017) estudiaron las propuestas de la Comisión de las Comunidades Europeas y de Australia y concluyeron que la identificación de IC requiere un enfoque estructurado, donde los gobiernos trabajen con los propietarios y operadores para priorizar la criticidad desde una perspectiva nacional. La identificación y designación de valores de criticidad generalmente se realiza de acuerdo con criterios nacionales predefinidos considerando los efectos cualitativos y cuantitativos de la interrupción o destrucción (parcial o total) de una infraestructura en particular. Siguiendo con el análisis de los autores, se propone la división de criterios en dos categorías:

- *Alcance*: la interrupción o destrucción de una infraestructura crítica en particular se clasifica según la extensión del área geográfica que podría verse afectada por su pérdida o falta de disponibilidad.
- *Gravedad*: las consecuencias de la interrupción o destrucción de una infraestructura en o la degradación de productos o servicios; número de población afectada; efectos ambientales y políticos.

Ante este panorama y dado el creciente interés de las distintas organizaciones terroristas por afectar las IC, el Consejo de Seguridad de las Naciones Unidas en su Resolución 2341/2017 hizo un llamamiento a los Estados miembros para que consideren desarrollar o mejorar aún más sus estrategias para reducir los riesgos a las IC de los ataques terroristas.

2.1.2 Sistemas críticos.

Dependiendo de la actividad que realizan los Sistemas, se pueden unificar los criterios del Departamento de Defensa de Estados Unidos y la OTAN como se presenta en el siguiente listado (Persano, M.L.,2015, p. 59):

- Sistemas relacionados con el uso de munición o con la precisión y disparo de montajes de armas, lanzadores y otros equipos;
- Sistemas de combate basados en computadoras, utilizados para calculo de datos críticos como los programas de análisis de resistencia, que trabajen con datos críticos para la seguridad o que supervisen otros sistemas por razones de seguridad;
- Sistemas que estén relacionados con fuentes de energía potencialmente peligrosas;

- Sistemas que controlen total o parcialmente el movimiento de un vehículo (aeronaves, vehículos terrestres, objetivos guiados por radar, etc.) o de piezas de equipos potencialmente peligrosas para las personas en sus proximidades.
- Cualquier unidad de software o módulo que se relacione con la detección de fallos, restauración de sistemas de seguridad, interrupciones programadas o de seguridad de procesamientos de datos, que controlen total o parcialmente hardware critico para la seguridad.

Los ataques que se producen a los Sistemas antes mencionados incluyen (Nickolov, E., 2005):

- Acceso no autorizado a información sensible o confidencial;
- Destrucción, modificación o sustitución del software que necesitan las Infraestructuras Críticas;
- Limitar el acceso a los agentes capaces de prevenir o mitigar los resultados de los ataques.

Otro tipo de ataque es aquel que puede quitar de funcionamiento un sistema mediante un ataque de DDoS (denegación de servicio distribuida). La investigación de Rehak, Senovsky y Slivkova (Rehak et al, 2018) demuestra que inicialmente el sistema puede utilizar sus reservas para procesar todas las solicitudes, pero si la intensidad de las nuevas solicitudes enviadas continúa aumentando, la proporción de solicitudes ilegítimas aumentará en consecuencia. Como resultado, el número de solicitudes legítimas procesadas seguirá disminuyendo hasta que el sistema ya no pueda responder. Sin embargo, una vez que disminuye la intensidad del ataque, el sistema comenzará a recuperarse gradualmente.

2.1.3 Desarrollo Seguro.

En comparación con las operaciones militares tradicionales, el espacio cibernético desafía las capacidades convencionales de los Estados con un costo relativamente bajo de operaciones, lo cual representa una ventana de oportunidad para los agresores (Rodríguez, 2017). Para llevar adelante dichas operaciones, se toma ventaja del avance de los paradigmas de programación y de los lenguajes que ellos contienen aprovechando su facilidad de integración, la velocidad y la practicidad. Asimismo, la evolución de los componentes y arquitectura del computador, así como también la de los estándares y normativas de calidad requieren un estudio particular que permita mejorar las capacidades defensivas y de contramedidas del Estado-Nación u Organización.

Una de las mayores debilidades de la revolución de las TIC es la creciente dependencia de los productos desarrollados por un número limitado de fabricantes. Es sabido que cuanto menor es la diversidad, mayor es el riesgo de un incidente de seguridad que puede generar un impacto catastrófico en todo el sistema. Un ejemplo es el caso de *GhostSecret*[3] en donde el atacante empleó múltiples herramientas para recopilar información de organizaciones específicas utilizando una vulnerabilidad de *Adobe Flash*, enviando correos electrónicos que tenía en adjunto un documento malicioso de Microsoft Word. Gran parte de estas vulnerabilidades pueden atribuirse por la calidad del desarrollo de software o malas configuraciones predeterminadas.

El CISSP establece que las "soluciones personalizadas pueden presentar grandes vulnerabilidades de seguridad como resultado de desarrolladores maliciosos y/o descuidados que crean trampillas, vulnerabilidades de desbordamiento del búfer u otras debilidades que pueden dejar un sistema abierto a la explotación por parte de personas malintencionadas." (Chapple, M. Gibson, D. y Stewart, J., 2015, p.838*)*

En 1999, el PITAC le informaba al presidente de los Estados Unidos la necesidad de fortalecer los procesos de producción de software de la siguiente forma:

> "La demanda de software ha crecido mucho más rápido que nuestra capacidad de producirlo. Además, la Nación necesita un software que sea mucho más usable, confiable y poderoso que lo que se está produciendo hoy. Nos hemos vuelto peligrosamente dependientes de grandes sistemas de software cuyo comportamiento no se entiende bien y que a menudo fallan de manera imprevista. Por lo tanto, los aumentos en la investigación sobre software deberían tener una alta prioridad. Se debe hacer especial hincapié en el desarrollo de software para administrar grandes cantidades de información, para hacer que las computadoras sean más fáciles de usar, para hacer que el software sea más fácil de crear y mantener, y para mejorar las formas en que los humanos interactúan con las computadoras." (PITAC, 1999, p.4)

Asimismo, remarcaba que resultaba fundamental financiar la investigación en métodos de desarrollo de software y tecnologías de componentes.

[3] Identificado con el código CVE-2018-4871 por la Organización MITRE. Para conocer más sobre este tema revisar: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-4871

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, 2015), considera que tener un software desarrollado a medida permite que la solución coincida exactamente con la necesidad de negocio que llevó a su producción. Además, el principal interesado puede monitorear el proyecto durante toda su ejecución y el mantenimiento, así como influenciar los cambios que sean pertinentes ya que el desarrollo estará bajo la órbita de la organización.

En esta línea el Plan de Protección de Infraestructuras Nacionales de Estados Unidos (NIPP, 2013) propone una metodología de planificación que describe el proceso para establecer metas y objetivos, identificar activos, sistemas y redes, evaluar riesgos, implementar programas de protección y estrategias de resiliencia, medir el desempeño y tomar medidas correctivas. Asimismo, remarca la importancia de compartir información a lo largo de todo el proceso con el objetivo de documentar y construir mejores prácticas basadas en las lecciones aprendidas.

En este sentido y con el objetivo de avanzar en el esfuerzo nacional de protección de la sociedad, se hace hincapié en la necesidad de promover la Investigación y Desarrollo (R+D) para permitir el diseño seguro y resistente y la construcción de infraestructura crítica y una tecnología cibernética de acompañamiento más segura (NIPP, 2013, p.25).

La Oficina de Responsabilidad del Gobierno de Estados Unidos (US GAO, 2017) presentó en el 2017 un resumen del análisis de cómo se desarrollaron los planes sectoriales durante el 2015, identificando riesgos emergentes en cada sector y lo presenta en la Tabla 1 en donde se puede observar que el Desarrollo Seguro es fundamental y común a casi todos los sectores identificados como IC.

Riesgos / Industria	Quimica	Instalaciones comerciales	Comunicaciones	Fabricación crítica	Represas	Servicios de emergencia	Energía	Servicios financieros	Alimentación y agricultura	Instalaciones gubernamentale	Servicios de Salud	Tecnologías de la Información	Nuclear	Transporte	Agua
Eventos climaticos más extremos y frecuentes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓
Crecientes ciberdependencias y ataques ciberneticos más sofisticados	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Amenazas crecientes de ataques terroristas internos y externos	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Envejecimiento de la Infraestructura por fondos limitados para reparaciones y mantenimiento			✓	✓	✓	✓	✓			✓	✓		✓		
Cadena de suministros cada vez más compleja y global	✓	✓	✓	✓							✓		✓		
Pandemias severas	✓	✓		✓						✓	✓				
Disponibilidad limitada de recursos para mejoras de resiliencia					✓	✓							✓	✓	✓
Interrupciones globales y sociopolíticas		✓	✓	✓			✓								

Tabla 1 Riesgos intersectoriales identificados durante la actualización del plan sectorial específico de 2015 (US GAO, 2017, p. 33)

Es decir, se debe incorporar lo aquí planteado en los sistemas de control industrial ya que son considerados como la columna vertebral de muchos sectores de la infraestructura críticos y se utilizan ampliamente en muchos campos de la industria, como el tratamiento eléctrico, de petróleo y de agua. Las ICs incluyen sistemas SCADA, de Control Distribuido y PLC (Al-Nashif, Y., 2013).

El análisis realizado por RTI (2002) demostró que los procesos de desarrollo tradicionales permiten que el *70% de las fallas se introduzcan tempranamente en el ciclo de vida*, mientras que el 80% de ellas no se detectan hasta la fase de prueba o aún más tarde cuando los costos de reparación son superiores a los costos al introducirlo. Otro análisis, realizado por Zaballos y Jeun (Jeun, I. y Zeballos A. G., 2016) para el Banco Interamericano de Desarrollo, utiliza como ejemplo el problema financiero que ocasionó el ataque a los sistemas de las ICs en septiembre del 2003. El análisis económico del impacto negativo del ataque para las infraestructuras interconectadas demostró que el sistema completo de cincuenta y seis industrias a nivel nacional tuvo pérdidas económicas de € 123,17 millones, y de € 81.79 millones para las once industrias de infraestructuras críticas (Dorneanu, et al., 2012).

La Figura 1 muestra las estadísticas publicadas por el NIST en donde se puede observar la variación de las vulnerabilidades más reportadas según la categorización sugerida por la organización Mitre[4].

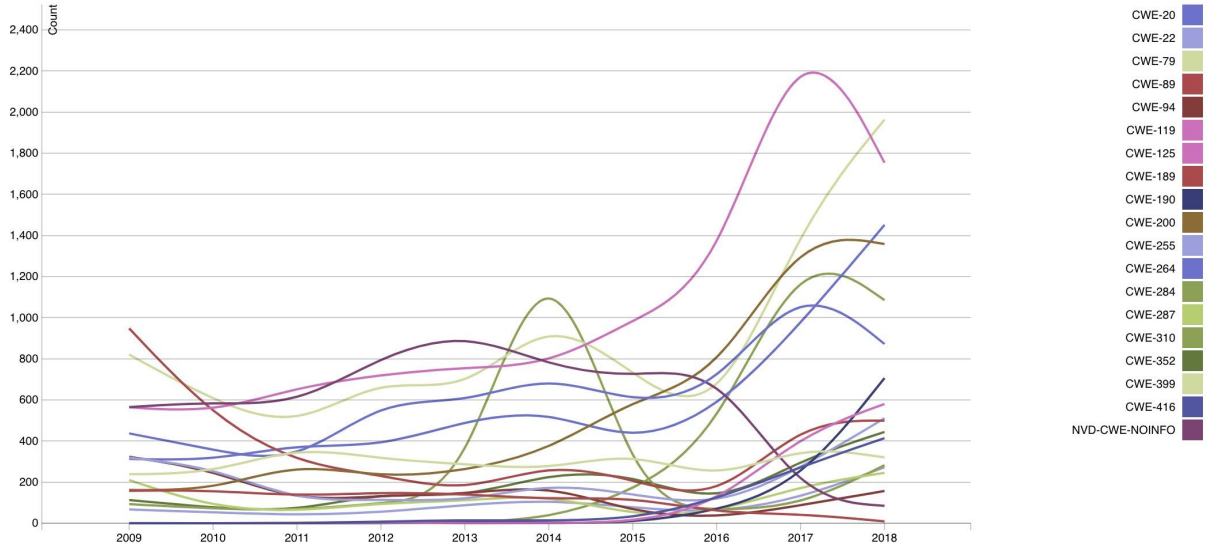


Figura 1 Vulnerabilidades por año (NIST, 2019b)

Las 10 vulnerabilidades más reportadas del último año son:

#	ID	Descripción
1	CWE-79	Neutralización incorrecta de las entradas durante la generación de la página web
2	CWE-119	Restricción inadecuada de operaciones dentro de los límites de un búfer de memoria
3	CWE-20	Validación de entradas incorrecta
4	CWE-200	Exposición de información
5	CWE-284	Control de acceso incorrecto
6	CWE-264	Permisos, privilegios y controles de acceso
7	CWE-190	Desbordamiento de enteros o envolvente
8	CWE-125	Lectura fuera de límites
9	CWE-89	Neutralización incorrecta de elementos especiales utilizados en un comando SQL
10	CWE-22	Limitación incorrecta de un nombre de ruta a un directorio restringido

Tabla 2 Vulnerabilidades por año (NIST, 2019b)

[4] Es una organización sin fines de lucro que opera múltiples centros de investigación y desarrollo financiados con fondos federales. Se dedica al estudio de la inteligencia artificial, la ciencia de datos intuitiva, la ciencia de la información cuántica, la informática de la salud, la seguridad espacial, la experiencia en políticas y economía, la autonomía confiable, el intercambio de amenazas y la resistencia cibernéticas. La sección dedicada al estudio de vulnerabilidades puede ser visitada en: https://cwe.mitre.org/index.html

La permanencia de vulnerabilidades tan comunes demuestra la necesidad de establecer un marco de desarrollo seguro común a todas las organizaciones y, fundamentalmente, en sistemas y aplicaciones de gran criticidad para una nación.

Shirazi (2009) concluye que la integración de la seguridad en los estándares procedimentales de desarrollo de software, junto con la falta de soporte para herramientas y métodos automatizados, son parte de los desafíos a los que se enfrenta la ingeniería de software en la actualidad.

2.2 Antecedentes de Normas, Estándares y Recomendaciones para el Desarrollo de aplicaciones en el marco de la ciberdefensa y ciberseguridad

2.2.1 Normas, Estándares y Recomendaciones.

El informe publicado en 2011 por el Departamento de Seguridad Nacional de los Estados Unidos clasifica en diversos sub-controles las actividades que se deben realizar para garantizar la seguridad de los sistemas (US. DHS, 2011). El trabajo realizado por Alcaraz y Zeadally propone una re-categorización y, en particular, define que el sub-control operativo

> "Comprende todos los sub-controles de seguridad que permiten que un sistema realice un conjunto de actividades (por ejemplo, control operativo o gestión de información confidencial) de forma segura. Dentro de esta clasificación, incluimos la adquisición de sistemas y servicios (por ejemplo, asignación o adquisición de activos, software y servicios del sistema de control), gestión de configuración, gestión de información y documentos, desarrollo y mantenimiento del sistema, protección del sistema y la comunicación, gestión y respuesta a incidentes, sistema e integridad de la información, control de acceso, auditoría y responsabilidad, y protección de los medios." (C. Alcaraz, and S. Zeadally. 2015, P. 16)

A continuación, se describen las normas, estándares y recomendaciones más utilizadas, en el **Anexo 2** se agregan tablas comparativas entre estos.

NIST 800-53

En la sección dedicada a nuevos desarrollos, se propone la incorporación de principios de control de seguridad en las etapas de desarrollo e implementación, así como también la utilización de un Marco de Gestión de Riesgos tanto para desarrollos nuevos como para sistemas heredados. (NIST, 2013)

NISTIR 7628

El conjunto de volúmenes de la norma presenta un marco analítico diseñado específicamente para el sector energético a favor de desarrollar estrategias efectivas de ciberseguridad y considerando las características propias de dicha industria a la hora de analizar riesgos y vulnerabilidades. (Smart Grid Interoperability Panel, 2010)

ISA 99-1 y 99-2

La serie ISA 99-1 incluye la descripción de conceptos y modelos, un glosario de términos y abreviaciones y las métricas de conformación de los sistemas de seguridad. En tanto en la 99-2, se describen las políticas y procedimientos que la organización debe seguir a la hora de crear y mantener un programa de seguridad. (INCIBE, 2015)

ISO 17799

La norma establece pautas y principios generales para iniciar, implementar, mantener y mejorar la gestión de la seguridad de la información en una organización. (ISO/IEC, 2005)

ISO 19791

La norma proporciona orientación y criterios para la evaluación de seguridad de los sistemas operativos. (ISO/IEC, 2015)

ISO 27001 e ISO 27002

Las normas se centran en el concepto de Sistema de Gestión de Seguridad de la Información y proporciona un modelo a fin de establecer, implementar, mantener y mejorarlo continuamente. La norma 27001 establece que dicho sistema protege la confidencialidad, integridad y disponibilidad de la información mediante un proceso de gestión de riesgos y proporciona confianza a las partes interesadas respecto a su correcta gestión. (ISO/IEC 27001 y ISO/IEC 27002, 2013).

IEC-62351

Es un estándar de la industria energética destinado a mejorar la seguridad de la información y cuyo objetivo principal es garantizar la integridad, autenticidad, confidencialidad y no repudio introduciendo mecanismos de autenticación (IEC 62351, 2018). Sin embargo, ha habido críticas a la norma como por ejemplo la planteada por Schlegel, Obermeier, y Schneider al afirmar que:

> "El estándar contiene algunas imprecisiones (por ejemplo, designaciones de conjuntos de cifrado) y opciones poco convencionales (por ejemplo, firmas RSA para IEC 61850). Tampoco considera los algoritmos criptográficos más nuevos que

podrían proporcionar las mismas garantías de seguridad a un costo de rendimiento más bajo (por ejemplo, criptografía de curva elíptica).” (Schlegel et al, 2015, p.18)

Quienes a su vez concluyen que en general, el “estándar proporciona un enfoque equilibrado que se puede implementar con un esfuerzo razonable y proporciona una cantidad razonable de seguridad si se implementa de manera integral”. (Schlegel et al, 2015, p.18)

FIPS 140-2 y FIPS PUB 140-3

La norma FIPS 140-2 especifica los requisitos de seguridad que cumplirá un módulo criptográfico utilizado dentro de un sistema de seguridad que protege la información confidencial pero no es clasificada (NIST, 2001). Mientras que la norma FIPS PUB 140-3, publicada en 2019, reemplaza la totalidad de los Requisitos de seguridad para módulos criptográficos cuyas áreas abarcan

> “La especificación del módulo criptográfico; interfaces de módulos criptográficos; roles, servicios y autenticación; seguridad de software / firmware; entorno operativo; seguridad física; seguridad no invasiva; gestión sensible de parámetros de seguridad; autocomprobaciones; aseguramiento del ciclo de vida; y mitigación de otros ataques” (NIST, 2019a, p. iii)

La norma está incluida en los Estándares de Firma Digital aprobados por la Secretaría de Modernización de la Jefatura de Gabinete de ministros de la República Argentina, y queda establecido como el protocolo para la protección de las claves privadas de certificadores y suscriptores.[5]

Wirelesshart, ISA100.11a y ZigBee

Son protocolos de comunicación basados en estándares como el IEEE 802.15.4 para cifrar comunicaciones en la capa de enlace utilizando esquemas de cifrado como AES-128, con especial importancia en comunicaciones como ser los sistemas **SCADA** presentes en la mayoría de la IC y de criticidad máxima.

AGA 12

La serie de recomendaciones AGA están diseñadas para garantizar la confidencialidad y autenticidad de las comunicaciones entre sistemas SCADA y están centradas en el enlace entre los servidores de comando y control y los dispositivos de campo.

[5] Para más información se recomienda visitar la página de la jefatura de gabinete de ministros de la República Argentina: https://www.argentina.gob.ar/modernizacion/administrativa/firmadigital/estandares

Familia de normas NERC CIP

Especifica los requisitos mínimos de seguridad para las compañías eléctricas. La norma 002 establece que se deben identificar ciberactivos críticos, la 003 propone crear políticas de seguridad para proteger esos activos y la norma 007 define los métodos, procesos y procedimientos para gestionar la seguridad de los sistemas dentro del perímetro.

GAO-04-140T

En el marco del estudio de la sensibilidad de las Infraestructuras críticas, la GAO se expidió respecto de un análisis de situación de las potenciales vulnerabilidades cibernéticas, cuyos resultados fueron publicados en el año 2013. El documento se centra en los riesgos asociados con los sistemas de control, ataques informáticos, claves para asegurar diversos sistemas de control y recomendaciones a seguir para fortalecer la seguridad de dichos sistemas tanto en el ámbito privado como federal. (GAO, 2003)

IEEE 1402

Es un estándar de seguridad física para las subestaciones de energía, en donde se propone una guía que identifica y analiza los problemas de seguridad relacionados con la intrusión humana. (IEEE, 2000 & US Dept. of Energy., 2005)

API Sec

Este estándar de seguridad SCADA proporciona orientación a los operadores de sistemas de tuberías de líquidos de petróleo y gas para gestionar la integridad y seguridad del sistema SCADA. (API, 2009).

2.2.2 Propuestas de integración de Normas.

Existen trabajos que proponen la integración de estándares en función de un sistema de gestión de procesos, en la Figura 2 se presenta el trabajo realizado por Mesquida, Más, San Feliu y Arcilla (2014) el cual explica que

> "Las intersecciones entre los elementos de la Figura son proporcionales a las relaciones detectadas entre los procesos de cada norma. Mediante esta representación se ha querido mostrar el esfuerzo aproximado que debería realizar una organización que ha iniciado un programa de mejora de procesos según la norma ISO/IEC 15504 para implantar las normas ISO/IEC 20000-1 e ISO/IEC 27001." (Mesquida et al., 2014, p.33)

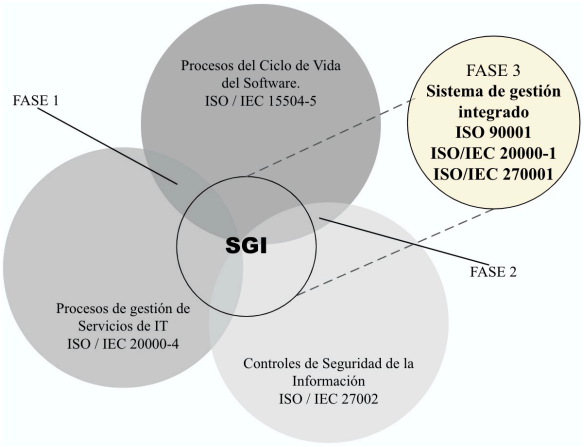


Figura 2 MIN-ITS: Marco Integrado de Estándares de gestión. (Mesquida et al., 2014, p.34)

Por último, el trabajo sostiene - al igual que lo presentado hasta aquí - que entre las normas existen varios elementos comunes (en particular analizan las coincidencias de las normas ISO 9001, ISO/IEC 20000-1 e ISO/IEC 27001). La ventaja de implementar un sistema de gestión que contemple varias normas permitiría reducir los recursos humanos, el presupuesto y tiempo necesarios para planificar, implementar y mantener dicho sistema. (Mesquida et al., 2014)

Nevada González (2014) analizó la combinación de normas y estándares que utilizan en el modelo Deming como línea de trabajo. En particular, de la combinación de ITIL V3 con ISO 20000 destacó que el resultado obtenido es la prestación de servicios de TI de calidad y facilidad a la hora de saber el grado de madurez de los procesos.

3. Aspectos específicos para tener en cuenta en el marco de ciberdefensa para el desarrollo seguro

3.1 Gestión del Ciclo de Vida en el Desarrollo de Software.

Según la ISO 12207, el Ciclo de Vida es un marco de referencia que contiene los procesos, actividades y tareas involucradas en el desarrollo, operación y mantenimiento de un producto de software y que abarca toda la vida del sistema desde la definición de sus requerimientos hasta el final de su uso.

Shirazi (2009) concluye que la integración de la seguridad en los estándares procedimentales de desarrollo de software, junto con la falta de soporte para herramientas y métodos automatizados, son parte de los desafíos a los que se enfrenta la ingeniería de software.

Lee y Shon (2016) afirman que la mayoría de los ataques cibernéticos[6] recientes a infraestructuras críticas son de Amenazas Avanzadas Persistentes (APT) bajo la explotación de vulnerabilidades de seguridad de *día-cero*, siendo necesario inspeccionar y complementar la infraestructura y el diseño de la seguridad a fin de evitar que estas sucedan.

Existen diferentes modelos en puja para ver cual es el más efectivo para organizar las diferentes actividades, técnicas, tareas de administración y de organización del ciclo de vida del desarrollo de software. Los ejemplos más difundidos son:

- Modelo en cascada (Royce, 1970);
- Modelo iterativo e incremental (Radatz, Olson, y Campbell, 1995);
- Modelo en espiral (Boehm, Penedo, Stuckle, Williams, y Pyster, 1984);
- Modelo unificado (IBM/Rational, 2001);
- Modelo ágil (Agile Methods and the Agile Manifesto).

Cualquiera sea el modelo aplicado, ya sea los anteriores u otros, no es necesario realizar cambios en sus pasos básicos a la hora de desarrollar aplicaciones y sistemas críticos.

Por lo general, **el ciclo de desarrollo será iterativo e incremental**, donde las piezas de software se irán acoplando e integrando con otras para crear componentes más robustos

[6] Ejemplos de estos ataques son el sufrido por la red de mensajería global SWIFT que utilizan las entidades bancarias para transferir dinero entre 2015 y 2016; el intento de toma de control remoto de la planta petroquímica Saudí Aramco en Arabia Saudita en agosto del 2017 y la penetración a la red y robo de datos a la planta de energía nuclear Kudankulam de India en septiembre del 2019.

para luego confluir en el producto terminado. Frente a ello, se debe integrar la seguridad en el Ciclo de Vida del Desarrollo de Software mediante la creación de un plan de seguridad para garantizar que cada actividad de seguridad se planee al comienzo del proyecto. (Hox & Boeije, 2005).

El marco de referencia debe ser independiente del tipo de sistema y el modelo de procesos y no debe interpretarse como una secuencia de flujos o pasos de procesos, sino más bien como un conjunto de contextos interactuantes cada uno con sus propios controles y equilibrios. El marco propuesto por el NIST se divide en tres contextos teniendo todos en común el análisis de seguridad del sistema en donde se incluye el ciclo de vida del proyecto como lo muestra la Figura 3. (Ross, Mcevilley, y Oren, 2018)

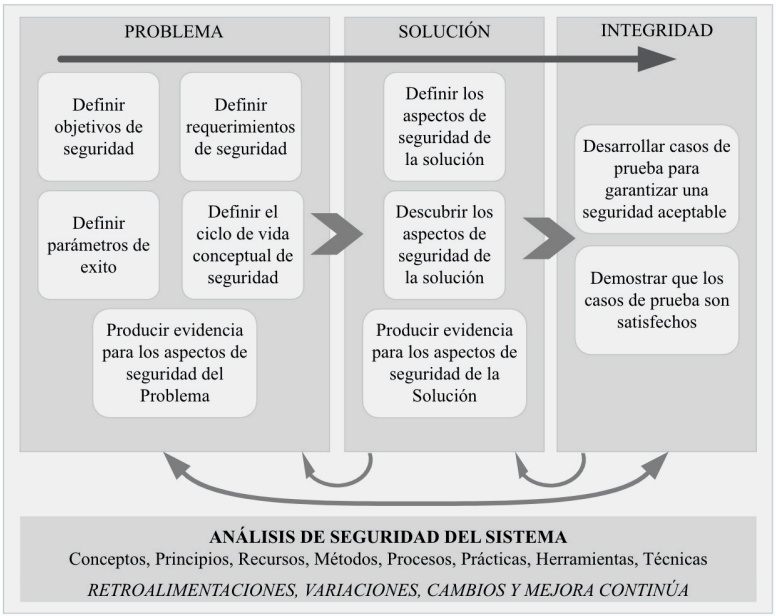


Figura 3 Marco de referencia de ingeniería de seguridad en Sistemas. (Ross et al., 2018)

El ciclo de vida propuesto por Jones y Rastogi (2004) propone una serie de actividades para cada fase del ciclo de vida como se puede ver en la Figura 4, incluida una fase destinada a la destrucción de elementos una vez que ya no son necesarios.

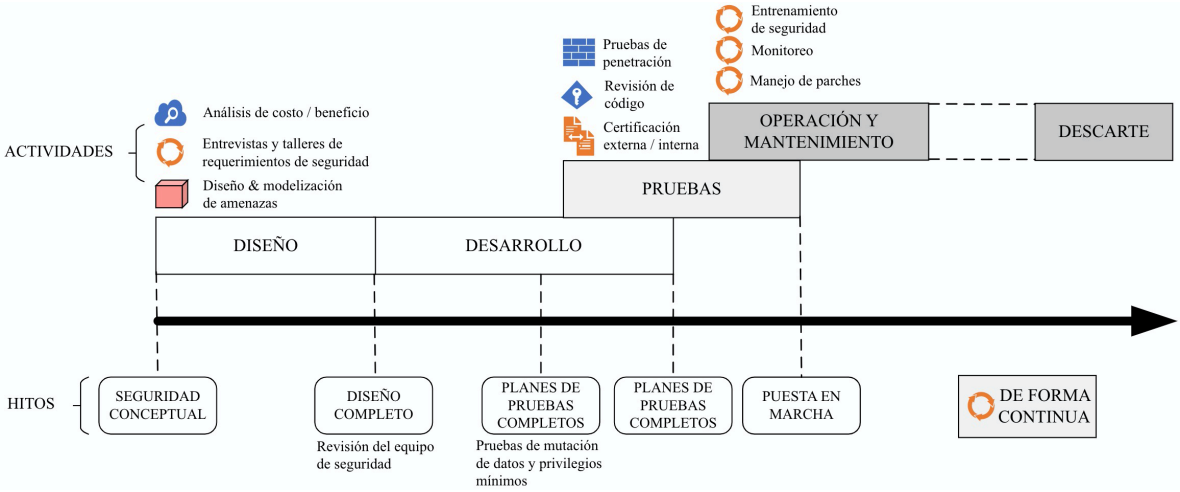


Figura 4 Ciclo de vida del desarrollo seguro de software. (Jones y Rastogi, 2004, p. 33)

El marco propuesto por el Instituto de Ingeniería de Software (*SEI*) tiene como estrategia principal la eliminación de defectos durante varias fases del ciclo de vida, proponiendo que a mayor cantidad de puntos de eliminación mayor probabilidad de encontrar defectos que se hayan introducido de forma temprana. La ventaja de esta estrategia es que, al detectar los defectos ni bien son introducidos, son más sencillos de remover y determinar cuando y donde se crearon.

Para Davis (2005), cada actividad de eliminación de defectos puede considerarse como un filtro que elimina cierto porcentaje de defectos que pueden generar vulnerabilidades (como ilustra la Tabla 3).

Etapa	Defectos eliminados
Requerimientos	Un porcentaje de vulnerabilidades inyectadas durante la etapa de requerimientos es eliminada durante el análisis de estos, el modelado de amenazadas o el desarrollo de casos de abuso.
Diseño	Un porcentaje de las vulnerabilidades inyectadas durante la etapa de requerimientos y la de diseño es eliminada durante las revisiones de diseño y las verificaciones del mismo.
Implementación	Un porcentaje de las vulnerabilidades inyectadas durante la etapa de requerimientos, diseño y codificación es eliminada durante las revisiones de código, los análisis dinámicos y estáticos y las pruebas de seguridad.
La meta de cada etapa será disminuir / eliminar los defectos producidos en las etapas anteriores y en la que se encuentra en ejecución.	

Tabla 3 Filtros de eliminación de vulnerabilidades. (Davis, 2005, p. 16)

Deloitte plantea que DevSecOps puede ayudar a mejorar los niveles de seguridad y madurez del cumplimiento en la secuencia de DevOps de una compañía, al tiempo que impulsan la calidad y la productividad y reducen el tiempo-al-mercado (Deloitte, 2019, p.103). En la Figura 5 se puede observar el enfoque planteado por DevSecOps, según el

estudio realizado por Deloitte, en donde se ve claramente qué actividades se incorporan en cada fase del ciclo de vida a fin de garantizar la seguridad a lo largo del desarrollo.

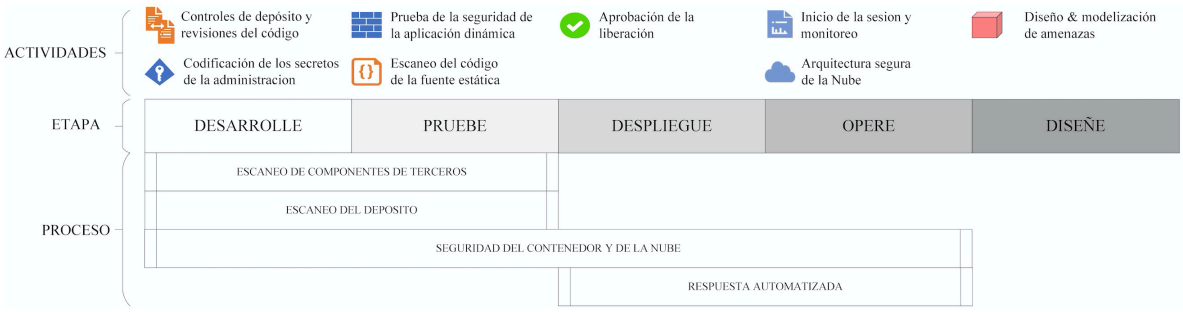


Figura 5. ¿Qué es DevSecOps? (Deloitte, 2019, p. 106)

Las organizaciones ISO/IEC/IEEE (2018) sugieren que el enfoque evolutivo puede ser aplicado a sistemas complejos y, en particular, a sistemas militares de tecnología de la información donde las nuevas capacidades desarrolladas pueden ser una modificación de algún bloque del producto existente o reemplazar directamente a la versión existente.

3.1.1 Especificación de Requerimientos

La especificación inadecuada de los requerimientos es uno de los principales contribuyentes al fracaso de un gran número de proyectos de software, independientemente de su tamaño (Donaldson y Jenkins, 2000). La investigación con base en muchos proyectos muestra que el 45% de las características propuestas en los requerimientos iniciales no se utilizaron, y un 19% adicional rara vez se utilizó. (Larman, 2004)

Devanbu y Stubblebine (2000) definieron que un requerimiento de seguridad es una manifestación de una política organizacional de alto nivel en los requisitos detallados de un sistema específico. (p.1) Los componentes funcionales de seguridad expresan requisitos destinados a contrarrestar las amenazas en el entorno operativo asumido. Por otra parte, la resiliencia describe la capacidad de detener, hacer frente, aclimatarse y / o recuperarse de incidentes que tienen consecuencias negativas. (US. DHS, 2013; Rehak et al., 2018)

Karim, Albuolayan, Saba, y Rehman (2016) proponen que en esta etapa se deben definir regulaciones aplicables, obtener los requerimientos de seguridad de todas las partes interesadas y adoptar estándares internacionales que se ajusten a la organización.

La evaluación de amenazas implica identificar y caracterizar con precisión los posibles ataques a fin de comprender mejor los riesgos y facilitar su gestión. Una falla de seguridad se define como el incumplimiento de los requerimientos, objetivos y medidas de desempeño relevantes para la seguridad, que incluyen exhibir comportamientos no

especificados, interacciones no especificadas o producir resultados no especificados. (Ross et al., 2018)

Castellaro, M., Romaniz, S., Ramos, J. y Pessolani, P. (2016) estudiaron distintas metodologías y encontraron que la mayoría recomiendan especificar el entorno operativo, identificar políticas globales de seguridad, roles y requerimientos, detallar casos de mal uso y ejecutar análisis de requerimientos de seguridad.

Desde la cultura DevSecOps, se plantea la necesidad de realizar trabajos pequeños los cuales, gradualmente, se cohesionan y hacen escalar el producto hasta el tamaño requerido. La *ventaja de tener esta la perspectiva* de trabajo es que es más fácil garantizar la seguridad de una pieza pequeña que la de un bloque de código de gran tamaño, obteniendo como resultado final un conjunto de piezas pequeñas seguras, lo que dará un bloque final seguro.

3.1.2 Diseño

Karim et al. (2016) proponen para esta etapa mantener una lista de los marcos de software recomendados, aplicar explícitamente principios de seguridad al diseño y obtener una revisión externa del diseño, mientras que Dawson et al. (2010) plantea desarrollar modelos de caso de uso y abuso, modelar amenazas y riesgos y realizar vistas previas contemplando un diseño seguro.

Castellaro et al. (2016) estudiaron distintas metodologías y encontraron que la mayoría recomiendan especificar propiedades de seguridad basadas en recursos, aplicar principios de seguridad al diseño, construir esquemas de etiquetado de la información, diseñar interfaces de usuario para funciones de seguridad y especificar configuraciones de seguridad de bases de datos.

Goertzel y Winograd (2008) remarcan que el *Common Criteria* propone niveles de garantía de evaluación (EAL) y que los requerimientos de seguridad para lograr los niveles más altos posibles de la práctica indican que el sistema debe estar diseñado y probado semi formalmente, este debe ser verificado semi formalmente y luego formalmente verificado y probado. Asimismo, remarcan que se deben mantener los datos del programa, los ejecutables y los datos de control y configuración del programa por separado, separando las entidades confiables de las entidades no confiables y minimizando el número de puntos de entrada y salida dentro y fuera de cualquier entidad.

Jones y Rastogi (2004) proponen el modelado de amenazas como actividad central de la fase de Diseño el cual contiene:

- Modelo de aplicaciones;
- Identificación de los nodos con mayor probabilidad de ser atacados al tener mayor superficie de ataque o portadores de información confidencial.
- Clasificación de amenazas para cada nodo identificado utilizando STRIDE (por las siglas en inglés de Suplantación, Manipulación, Repudio, Divulgación de información y Denegación de servicio);
- Un árbol de amenazas que contenga todas las vulnerabilidades detectadas.

Siguiendo con el último punto, la organización SAFEcode (2018) remarca que el modelado de amenazas es una práctica fundamental de la fase de diseño y debe realizarse lo antes posible en el ciclo de vida del producto, en donde el contexto generado por este modelado servirá para enfocar el análisis estático y las pruebas de seguridad.

Para la comunidad DevSecOps la fase de Diseño es la parte central de la seguridad, ya que desde aquí se pueden incorporar diversos principios al diseño del código, con el posterior ahorro de tiempo y dinero.

Humphrey (2000) propone usar las pautas de planificación de calidad del TSP (Ver **Anexo 1**) cuando no se cuenten con datos históricos de inyección de defectos, los cuales son fundamentales para elaborar el plan de calidad a seguir por el Estado durante el desarrollo de un nuevo producto.

Castellaro et al. (2016) estudiaron distintas metodologías y encontraron que la mayoría recomiendan revisar seguridad a nivel de código, conducir pruebas de fallas inducidas, integrar análisis de seguridad en el proceso de construcción y construir guías de seguridad operacional.

Goertzel y Winograd (2008) plantean que se deben considerar cuestiones relacionadas a la arquitectura del entorno de desarrollo para evitar la introducción de vulnerabilidades y/o código malicioso, tales como firewalls a nivel de aplicación, *cajas de arena* para probar los componentes de forma aislada y validadores de firma de código.

3.1.3 Pruebas

La producción de software seguro depende, en gran parte, del resultado de las pruebas que se realicen para validar que se han cumplido los requerimientos de seguridad, la correcta implementación de contramedidas y la producción de código de calidad. El trabajo realizado

por Jones y Rastogi (2004) enfatiza en este punto y destaca que la correcta verificación del diseño del sistema y el código es necesaria para determinar si son capaces de resistir un ataque. Asimismo, los Estados y las organizaciones deben ser cautelosas al incorporar o consultar cada pieza de software ya que podrían estar contaminadas debiendo realizar pruebas y evaluaciones en profundidad después de la adquisición o desarrollo y antes de la implementación o integración con el sistema en su conjunto. (NCIIPC, 2015)

Para Karim et al. (2016) se debe contar con un plan de pruebas de seguridad, realizar pruebas de penetración, utilizar herramientas de análisis estático y tener definido el nivel de aceptación de vulnerabilidades dentro de la etapa de codificación.

Castellaro et al. (2016) estudiaron distintas metodologías y encontraron que la mayoría recomiendan conducir pruebas de usabilidad de funciones de seguridad, identificar e implementar pruebas de seguridad y verificar atributos de seguridad de recursos.

De igual forma, SAFEcode (2018) considera que son útiles para validar tanto la efectividad de las prácticas realizadas como para detectar las fallas que se hayan introducido, para esto recomiendan utilizar herramientas de análisis estático y dinámico, analizadores de comportamiento ante mutación de datos (*fuzzing*), escaneos de vulnerabilidades y pruebas de penetración.

3.1.4 Implementación y Mantenimiento

Dawson et al. (2010) proponen realizar un despliegue seguro en una máquina virtual reforzada, realizar una evaluación de vulnerabilidades, una prueba de escaneo de superficie y ajustar el riesgo. Una vez realizada la implementación, es importante asegurar el entorno operativo. Es decir, que es recomendable agregar dispositivos externos como sistemas de detección de intrusos (si es que no se incluye en el producto desarrollado) y firewalls.

El marco de referencia de Desarrollo SAMM de OWASP remarca en su función de negocio *Gestión de Problemas* la práctica de **Endurecimiento Ambiental** para establecer un proceso de gestión del mantenimiento del entorno de operaciones y supervisar el estado de la configuración de la línea de base.

Del conjunto de actividades que propone el NIST (Ross et al., 2018) en está etapa se destacan:

- Establecer y controlar que las interconexiones del sistema desarrollado con su entorno se realicen de la forma en la que fue planificado y sean seguras.

- Definir la estrategia de mantenimiento y entregarla juntamente con el sistema desarrollado, la cual debe contener aspectos relacionados al mantenimiento correctivo, preventivo, de logística, de falsificación de identidades, capacidad de reemplazo preventivo o total de elementos ante fallas aleatorias o vulneraciones de la seguridad.

Para Haridas (2007) se debe definir un equipo que certifique el proceso y crear una fuerza de tareas que ayude a encontrar potenciales problemas que puedan detener el proyecto.

Para la comunidad DevSecOps remarca que las pruebas de penetración deben servir para la verificación puntual del diseño y la implementación, y como retroalimentación para la mejora de la organización.

3.2 El concepto de Privacidad por Defecto (PbD).

Es el conjunto de medidas tecnológicas que se emplean para garantizar la privacidad de forma **proactiva**, se encuentra el considerar los requisitos de privacidad a partir de la fase de diseño y a lo largo de todo el ciclo de vida de los datos, y **preventiva,** el no esperar que sucedan violaciones de estas reglas para tomar medidas sino tenerlas definidas previamente por defecto (Cavoukian, A., 2014).

Una forma de garantizar esto es que la configuración de privacidad de los sistemas y aplicaciones esté brindada por defecto y sin la necesidad de realizar configuraciones adicionales a favor de la protección de los datos personales. En dicho contexto, el usuario no debería realizar ninguna acción para proteger su privacidad ya que se encuentra integrada en los sistemas que utiliza y las operaciones que realiza. En este caso, la arquitectura debe ser diseñada de forma holística, integradora y creativa. Holística, porque siempre se deben considerar contextos adicionales más amplios. Integradora, porque todos los interesados deben ser consultados. Creativa, porque incorporar privacidad a veces significa reinventar las opciones existentes porque las alternativas (en general parches) son inaceptables.

Otro punto por considerar es *el enfoque de gestión de riesgos con miras a escoger e implementar medidas para una protección efectiva.* Los activos que proteger son las personas cuyos datos se procesan y, en particular, sus derechos y libertades fundamentales. Además, una vez finalizada la transacción se debe destruir el mayor volumen de datos posible, limitando aquellos que sean necesarios y garantizando el anonimato transaccional al máximo.

La medidas y criterios asociados a la privacidad deben implementarse en el diseño de un sistema o software para que las organizaciones no deban lidiar con problemas de

privacidad después del desarrollo. Esto se corresponde a la necesidad de minimizar los costos de mantenimiento de los sistemas y de prever los cambios futuros que puedan tener los requerimientos legales, a sabiendas de los elevados costos que conlleva modificar un sistema y las consecuencias que esto tiene. Ejemplo, es más fácil incorporar en la etapa de diseño medidas preventivas a la violación de datos personales que afrontar el costo de una demanda legal después de ocurrido dicho hecho.

A la fecha se han llevado a cabo diversos esfuerzos de estandarización para integrar los requisitos de privacidad en los sistemas. Por ejemplo, la Organización Internacional de Normalización (ISO por sus nombres en inglés) ha emitido un marco de referencia para privacidad (ISO / IEC 29100) y una arquitectura de privacidad (ISO / IEC 29101) relacionada con la Información de Identificación Personal (PII por sus siglas en inglés) dentro de un entorno de tecnología de información y comunicación. También extendió las normas ISO / IEC 27001 y 27002 sobre gestión de seguridad de la información a gestión de privacidad.

Esta metodología se vincula con el artículo 25 del Reglamento General de Protección de Datos (GDPR), afirma que se deben tomar medidas de protección desde la fase de diseño hasta la implementación y mantenimiento, apuntando a todo el ciclo de vida del proyecto e identificando la protección de datos personales dentro de los requerimientos del sistema. Si bien el GDPR no impone las medidas vinculadas a la gestión de riesgos a ser tomadas en consideración, propone como factores de estos el análisis de la naturaleza, el alcance, contexto y propósitos del procesamiento de datos.

Una limitación grave es que se aplica solo para *imponer una obligación a los controladores y no a los desarrolladores* ni a la tecnología utilizada para procesar datos personales.

3.2.1 Ejemplos de metodologías y recomendaciones existentes

Instituto de Ingenieros Eléctricos y Electrónicos

En el documento *Objetivos de protección para la privacidad en Ingeniería* (Hansen, M. Jensen, M. y Rost, M., 2015), se describe un marco para identificar los puntos a proteger en los sistemas que procesan datos personales. Allí se menciona la triada de la seguridad de la información y se le agregan tres objetivos adicionales: No vincularidad, Intervencibilidad y Transparencia como se representa en la Figura 6.

La primera se refiere a la capacidad de las piezas de información de ser anónimas, es decir, de estar relacionadas entre sí y a un sujeto. La segunda permite la aplicación efectiva

de los cambios y las medidas correctivas siendo esto relevante para dar cumplimiento a los derechos de las personas y la posible intervención de las autoridades competentes. Por último, el objetivo primario de la transparencia define que todos los procesos de información relativos a privacidad (incluido el entorno legal, técnico y organizativo) debe ser comprendido y reconstruido en cualquier momento. (Hansen, M. Jensen, M. y Rost, M., 2015),

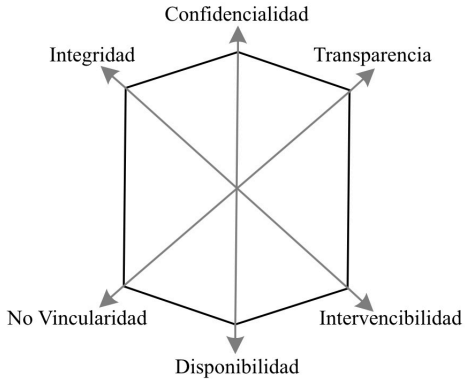


Figura 6. Los seis objetivos de protección para la ingeniería de privacidad. (Hansen, M. Jensen, M. y Rost, M., 2015)

Instituto Nacional de Estándares y Tecnología de los Estados Unidos (National Institute of Standards and Technology)

El NIST ha adoptado la definición de ingeniería de privacidad (NIST, 2017) como una disciplina especializada de ingeniería de sistemas enfocada en lograr liberarse de las condiciones que pueden crear problemas para las personas con consecuencias inaceptables que surgen del sistema mientras procesa Identificadores Personales de Identidad (PII[7]). Aquí se considera que la ingeniería de privacidad está compuesta por muchos componentes que tienen como troncales principales la gestión de riesgos y la triada de la seguridad de información, y agrega como objetivos fundamentales la previsibilidad, la capacidad de gestión y la disociabilidad.

[7] El NIST lo define como cualquier información que pueda usarse para distinguir o rastrear la identidad de un individuo, como nombre, número de seguro social, fecha y lugar de nacimiento, apellido de soltera de la madre o registros biométricos; y cualquier otra información vinculada o vinculable a un individuo, como información médica, educativa, financiera y laboral.

Universidad Católica de Lovaina

La metodología LINDDUN[8] propuesta por la Universidad Católica de Lovaina enfatiza en la dimensión de análisis de riesgos de la siguiente forma:

Figura 7.Pasos de la metodología LINDDUN (KU Leuven University, 2014).

Asociación de auditoría y control de sistemas de información (ISACA)

ISACA entiende que la privacidad es el derecho de una persona a confiar en que otros usarán, almacenarán, compartirán y eliminarán de forma adecuada y respetuosa su información personal, sensible dentro del contexto, y de acuerdo con los propósitos para los cuales fue recolectada u obtenida (ISACA, 2016).

3.2.2 Consideraciones legales

A la hora de diseñar un sistema, no solo se deben delinear los requerimientos de privacidad, sino que también se debe poder demostrar su cumplimiento. Durante todo el proceso de diseño se debe seguir la estrategia adoptada, la cual debe ser homogénea durante todo el ciclo de vida del producto. Estas estrategias no sólo son relevantes cuando se desarrollan sistemas propios, sino que deben ser tenidas en cuenta cuando se arman pliegos de licitación o detalles técnicos de sistemas a adquirir.

Un punto fundamental que se debe evaluar es si las ideas a desarrollar pueden ser ejecutadas dentro de los límites legales a los que estará sujeta nuestra aplicación, esto no es un detalle menor, especialmente para evitar conflictos entre Estados. En esta línea, la Agencia

[8] Del ingles: *Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Un- awareness, y Non-compliance.*

Española de Protección de Datos (AEPD, 2018) define a las Evaluaciones de Impacto de Privacidad como:

> "Un proceso ligado a los principios de protección de datos desde el diseño y protección de datos por defecto concebido para describir, de manera anticipada y preventiva, un tratamiento de datos personales, evaluar su necesidad y proporcionalidad y gestionar los potenciales riesgos para los derechos y libertades a los que estarán expuestos los datos personales en función de las actividades de tratamiento que se lleven a cabo con los mismos, determinando las medidas necesarias para reducirlos hasta un nivel de riesgo aceptable." (AEPD, 2018)

Algunas de las normas asociadas son:

El artículo 25 del GDPR trata sobre la privacidad por diseño.

Las leyes CalOPPA[9] y CCPA[10] de los Estados Unidos obligan a las empresas a la protección de datos.

3.3 El método de corrección por construcción (CbC).

Esta metodología (Chapman, R. Hall, A., 2002) promueve la alta integridad para aplicaciones de seguridad crítica[11] teniendo, como principios fundamentales, la no inclusión de errores, la eliminación de errores lo más rápido posible desde su introducción y la producción de entregables en cada proceso respaldados por diversas actividades. Las recomendaciones generales son:

1. Usar una notación sólida y formal para todos los entregables.
2. Usar métodos sólidos y compatibles con herramientas para validar cada entregable.
3. Realizar pequeños pasos y validar cada entrega.

[9] Ley de protección de la privacidad en línea de California (CalOPPA) fue la primer Ley estatal de los Estados Unidos de Norteamérica en requerir a los sitios web comerciales y de servicios en línea la publicación de sus políticas de privacidad entró en vigencia en el año 2004.

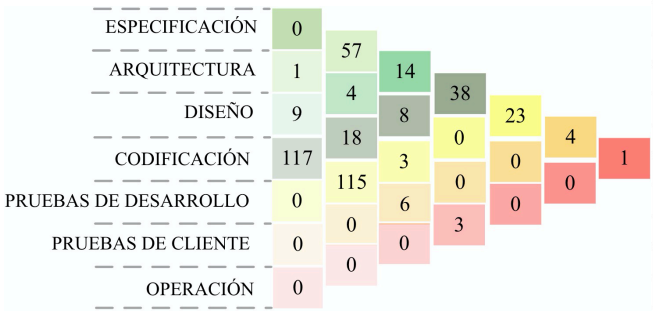
[10] Ley de privacidad del consumidor de California (CCPA) otorga a los consumidores nuevos derechos con respecto a la recolección de sus datos personales, entró en vigencia a fines del año 2018.

[11] La metodología hace distinción entre *security* y *safety*. En inglés el primer término es usado para referirse a amenazas externas que están orientadas a infligir daño a un individuo, organización o incluso activos. y safety para aspectos internos en donde uno tiene el control del riesgo que causa aspectos, por lo tanto, se protege contra riesgos que no son intencionados.

4. La información que se encuentra en la especificación no debe repetirse en el diseño y así en cada etapa.
5. El diseño del Software debe ser fácil de validar.
6. Construir prototipos en cada etapa para abordar los potenciales problemas al principio del proyecto.

Como resultado de las recomendaciones, se logra que los defectos sean rápidamente detectados generando cambios sencillos y de bajo costo, validación del correcto funcionamiento del Software en las etapas de pruebas y se omite la depuración de errores encontrados en etapas posteriores. Ya que desde un comienzo las iteraciones generan un producto útil, la confianza general del proyecto crece exponencialmente.

La Figura 8, extraída del trabajo presentado por Chapman y Hall (2002), muestra que en las fases del ciclo de vida en donde se introducen, detectan y remueven los errores en los desarrollos de software, siguiendo la metodología *CbC* los valores obtenidos y representados por el lado derecho deberían tender a cero. Asimismo, la distribución del esfuerzo aplicando la metodología expone que la corrección de errores representa un porcentaje relativamente bajo del proyecto. Esto es una gran ventaja respecto de la situación que se encuentra normalmente en los proyectos de sistemas críticos, donde las fallas suelen ser detectadas en etapas maduras y el costo de rediseñar, modificar e implementar las mejoras se eleva considerablemente.



Actividad	Esfuerzo (%)
Requerimientos	2
Especificación y Arquitectura	25
Codificación	14
Pruebas	34
Arreglo de fallas	6
Administración del Proyecto	10
Entrenamiento	3
Diseño de autoridad	3
Entornos de desarrollo y destino	3

Figura 8. Del lado izquierdo las fases del Ciclo de Vida con los errores y del derecho la tabla de distribución del esfuerzo (Chapman, R. y Hall, A., 2002, p. 24)

La metodología resulta de gran utilidad en sistemas complejos en donde la eficiencia es un pilar de los requerimientos (Cleophas, L., Kourie, D. G. y Watson, B.W., 2015). En comparación al Modelo de Madurez de Capacidades[12], la tasa de defectos por líneas de

[12] *El Modelo de Madurez de Capacidades proporciona a las organizaciones de software el modelo de referencia necesario como soporte para el control de sus procesos de desarrollo y mantenimiento y para*

código fuente (SLC) fue de $\frac{0{,}1\ defecto}{1000\ SLOC}$ contra $\frac{1\ defecto}{1000\ SLOC}$ de las organizaciones con un nivel 5 de CMM, y si bien es cierto que existen otras metodologías con tasas de defecto bajas (que se producen por pruebas extensas y costosas, seguidas de procesos de depuración), no suele ofrecer una tasa de productividad como CbC. En la Figura 9 se muestra una comparación contra los distintos niveles realizada por el estudio antes mencionado:

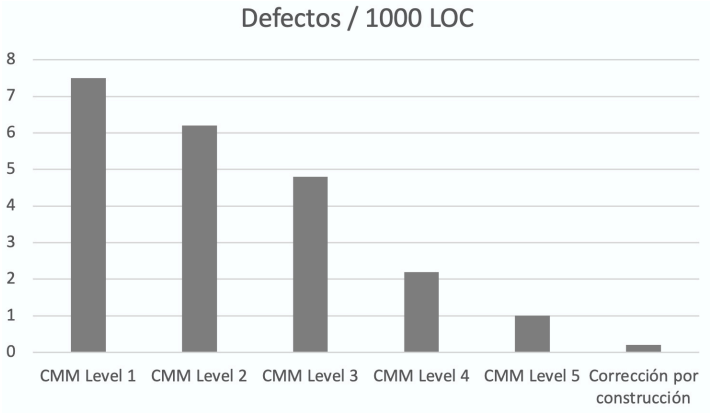


Figura 9. Tasas de defectos comparadas entre CbC y CMM. (Capers, J., 2000, p. 5)

Este método presenta algunas limitantes que deben ser consideradas a la hora de aplicarlo en una Organización. La primera que tiene la metodología es la propia cultura organizacional: se debe creer que es posible tener tanto una alta tasa de productividad como una muy baja tasa de defectos. Si bien esto no parece un problema a simple vista, la adopción de nuevas prácticas de trabajo de profesionales y en organizaciones que desarrollan software no es una tarea sencilla, y la hora de aplicar cambios de paradigma en el Estado, está limitación se incrementa aún más. Otro limitante mencionado por la literatura es que al ser una metodología tan rigurosa y precisa se requiere una fuerte motivación y un gran liderazgo por los patrocinadores de los proyectos y la alta gerencia en todos los proyectos.

3.4 Los conceptos de Seguridad por Defecto y por Diseño.

En 1974 Jerome Saltzer y Michael Schroeder propusieron por primera vez el concepto de Seguridad por Diseño en su artículo "La Protección de la información en los Sistemas de computadora". Acorde al Foro de Garantía de Software para la Excelencia en el Código (SAFECode, 2018), los principios para el diseño de sistemas seguros que han trascendido durante los últimos años son:

facilitar su evolución hacia una cultura de la Ingeniería del Software y de excelencia en la gestión. (Caballero, I., García O. y Piattini G., 2007, p. 158).

- Economía del mecanismo: mantener el diseño del sistema lo más simple y pequeño posible.
- Valores predeterminados a prueba de fallos: basar las decisiones de acceso en el permiso en lugar de exclusión.
- Mediación completa: verificar el acceso a cada objeto para su autorización.
- Privilegio mínimo: operar todos los programas y usuarios del sistema con la menor cantidad de privilegios necesarios para completar el trabajo.
- Mecanismo común mínimo: minimizar la cantidad de mecanismos a más de un usuario y dependiente de todos los usuarios.
- Aceptabilidad psicológica: diseñar la interfaz para facilitar su uso, de modo que los usuarios apliquen correctamente los mecanismos de protección de forma rutinaria y automática;

En la actualidad, un número cada vez mayor de países incorpora conceptos de Seguridad por Defecto en sus estrategias para aumentar la resistencia de las IC frente a ataques terroristas. La recomendación propone abordar un enfoque holístico con el fin de resolver los problemas de raíz en lugar de tratar los síntomas. Se trata así de pensar a largo plazo, garantizando desde el inicio del proyecto que las medidas de seguridad se encuentren integradas desde el software de base.

La Estrategia de Seguridad Cibernética de Singapur (Cyber Security Agency of Singapore, 2016) establece específicamente que el Gobierno se ha comprometido a tomar las siguientes medidas:

- Promover la práctica de las pruebas de penetración para descubrir vulnerabilidades en forma temprana para la corrección en la etapa de diseño.
- Construir una comunidad de práctica sólida en pruebas de productos y sistemas basadas en estándares internacionales establecidos, como la certificación de garantía de productos Common Criteria (CC);
- Continuar perfeccionando las metodologías y desarrollar nuevas herramientas de validación de seguridad para mejorar la eficacia de Seguridad por Diseño.

Es extremadamente importante considerar las implicaciones de seguridad de un proyecto de desarrollo de software desde las primeras etapas porque es mucho más fácil incorporar seguridad a un sistema que agregar seguridad a un sistema existente. (Chapple, M. Gibson, D. y Stewart, J., 2015).

3.4.1 Recomendaciones existentes

Centro Nacional de Seguridad Cibernética

Los principios recomendados por el NCSC (NCSC, 2018) implican que la seguridad debe integrarse desde el principio para tratar la causa raíz de un problema no esperando a que este se manifieste. Esta debe evolucionar constantemente para enfrentar la constante actualización de los atacantes ya que *las nuevas características de seguridad deben tomar más tiempo para vencer que para construir.*

OWASP

El proyecto OWASP propone en su recomendación Principios de Seguridad por Diseño (OWASP, 2016) los siguientes principios:

1. Minimizar la superficie de ataque: Se busca reducir los campos expuestos y el comportamiento que ellos tienen, por ejemplo, limitando las funciones de búsqueda a usuarios autorizados o utilizando rutinas de validación antes de realizar acciones.
2. Establecer valores predeterminados seguros: El formato predeterminado en el que se entrega un producto debe ser seguro, y solo en casos particulares se debe permitir al usuario realizar algún tipo de modificación sobre estos valores.
3. Principio del mínimo privilegio: Recomienda que las cuentas tengan la menor cantidad de privilegios para realizar sus procesos comerciales. Esto abarca los derechos de usuario, los permisos de recursos, como los límites de CPU, la memoria, la red y los permisos del sistema de archivos.
4. Defensa en profundidad: Los controles, cuando se usan en profundidad, pueden hacer que las vulnerabilidades severas sean extraordinariamente difíciles de explotar y, por lo tanto, es poco probable que ocurran.
5. Fallar de forma segura: Cuando ocurren problemas en las transacciones, puede ocurrir que se brinde más información de la necesaria. Cómo fallan puede determinar si una aplicación es segura o no.
6. No confíes en los servicios: No se debe confiar implícitamente en servicios que se consuman de proveedores externos, estos deben recibir el mismo tratamiento de seguridad que los desarrollos propios.
7. Separación de tareas: Establecer jerarquías en las funciones y en los clientes habilitados para invocarlas es una de las prácticas más utilizadas.

8. Evite la seguridad por oscuridad / ofuscación: Basar la estructura de seguridad en esconder la información es un control de seguridad débil, y los sistemas y funciones más relevantes no deben depender de ello.
9. Mantenga la seguridad sencilla: Fuertemente relacionado con el principio de la superficie de ataque, no es recomendable utilizar arquitecturas muy complejas si existen alternativas más simples.
10. Solucione los problemas de seguridad correctamente: Una vez que se ha identificado un problema de seguridad, es importante desarrollar una prueba para él y comprender la causa raíz del problema.

Propuesta de Ciclo de Vida basado en Seguridad por Diseño del CSA de Singapur

Aquí se propone que garantizar **un desarrollo eficiente** suele ser más preponderante que garantizar que sea seguro, con lo cual las mejoras de seguridad (que por lo general vienen de requerimientos de la etapa de testeo del producto) son introducidas en etapas posteriores al desarrollo. Al igual que lo que sostiene el presente trabajo esta metodología suele ser efectiva, aunque trae demoras y costos elevados. Es por ello que la Agencia de Ciberseguridad de Singapur (CSA of Singapur, 2017) ha propuesto una comparativa entre su proyecto de Ciclo de Vida (Figura 10) y el de Desarrollo de Sistemas tradicional *(SDLC)* extendiendo a todas las fases la identificación de riesgos a la seguridad mediante los siguientes pasos:

A. Cambiar los requisitos o la implementación para evitar el riesgo de seguridad;
B. Implementación de controles alternativos o atenuantes;
C. Aceptar el riesgo mediante un proceso de gestión de riesgos adecuado;
D. Procesos iterativos donde la seguridad se evalúa en cada fase y se determina si los procesos de seguridad deben repetirse para producir un resultado satisfactorio.

Inclusive y a la par del presente trabajo, la propuesta del CSA remarca que la ventaja de introducir seguridad junto con cada fase de SDLC es garantizar que los riesgos de seguridad sean visibles, bien entendidos por la alta gerencia y el personal clave, y que las decisiones apropiadas se tomen a tiempo para reducir el riesgo a un nivel aceptable. (CSA of Singapur, 2017)

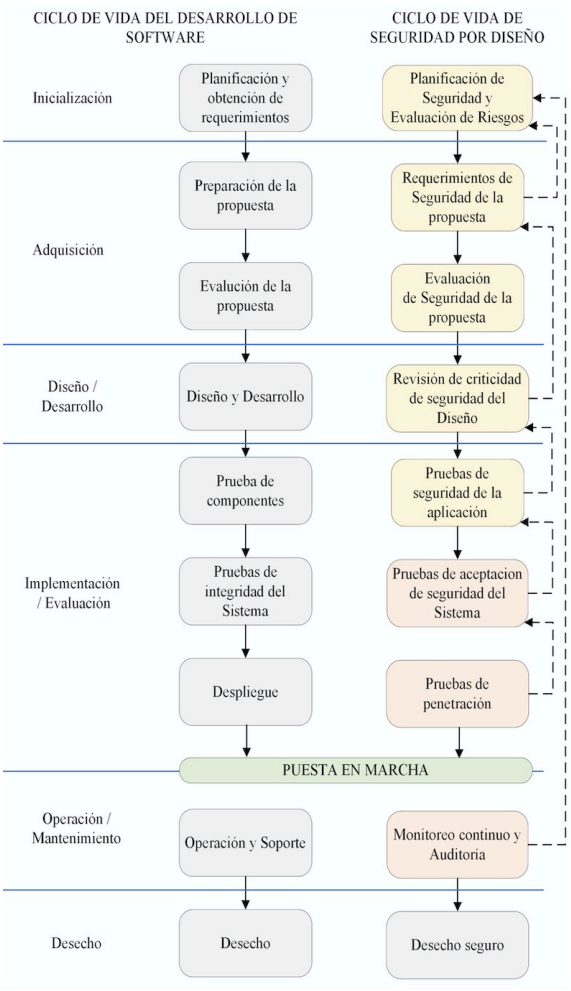


Figura 10. Ciclo de vida del desarrollo de software comparado con el de Seguridad por Diseño (Cyber Security Agency of Singapore, 2017)

3.5 Calidad en el Desarrollo de Software.

La IEEE (IEEE, 1998) define la calidad del software como el grado en que un software posee la combinación deseada de atributos de calidad del software.

Colon et al. (2007) proponen una definición dentro del contexto de la presente tesis:

> "La base para obtener una confianza justificable de que el software exhibirá de manera consistente todas las propiedades requeridas para garantizar que el software, en operación, continúe operando de manera confiable a pesar de la presencia de fallas patrocinadas (intencionales). En términos prácticos, dicho software debe ser capaz de resistir la mayoría de los ataques, tolerar tantos ataques como sea posible, y contener el daño y recuperarse a un nivel normal de operación lo antes posible después de cualquier ataque que no pueda resistir o tolerar." (p. 20)

Los atributos de calidad se han identificado en muchos estándares y modelos y, al ser cualitativos, se requiere de métricas para cuantificarlos. Existen diversos trabajos que proponen herramientas de evaluaciones automáticas de la calidad de los códigos fuente con el uso de diversas métricas. Como ejemplo, Fernando et al. (2017) generaron una herramienta de medición llamada SQAT utilizando el enfoque GQM (Goal Question Metric) para calcular los puntajes de los atributos de calidad de un software que demostró ser altamente eficiente, escalable y adaptable a diversos proyectos.

Haralambiev, Boychev, Lilov y Kraichev (2011) afirma que algunas de las prácticas de calidad deben ser incorporadas desde el inicio de los proyectos para que el diseño de calidad que se propone implementar se pueda verificar continuamente, abarcando los entornos de integración, la revisión de código, etc. Asimismo, aclaran que resulta beneficioso incorporar herramientas de análisis de calidad de código a fin de prevenir futuras vulnerabilidades. Adicionalmente, Steidl, Deissenboeck, Poehlmann, Heinke, Uhink-Mergenthaler (2014) concluyen que los análisis de calidad no deben basarse únicamente en mediciones automatizadas, sino que deben combinarse con una cantidad significativa de evaluación e interacción humana, ya que de está forma se puede garantizar que el sistema siga cumpliendo con las metas de calidad.

A la hora de establecer parámetros comunes para medir la calidad de los productos (abaratando los costos de control), se han creado certificaciones de estándares que las empresas utilizan a fin de garantizar calidad en el producto o servicio que ofrecen.

3.5.1 Normas y Recomendaciones asociadas

La norma *ISO 9001:2015* remarca que los potenciales beneficios que puede obtener una organización al implementar un Sistema de Gestión de la Calidad (SGC) involucran la capacidad de cumplir y demostrar los requisitos (ya sea legales, de funcionamiento, etc.) del cliente de forma regular aumentando así el grado de satisfacción de este, abordar riesgos y oportunidades teniendo un panorama claro del contexto y los objetivos del negocio.

La familia de normas *ISO/IEC 25000* sustituyeron a las normas ISO/IEC 9126 e ISO/IEC 14598 y proponen un marco para evaluar la calidad del producto de software. Dichas normas contemplan tres métricas de calidad a ser evaluadas: las internas (como el código fuente), las externas (como el comportamiento en la etapa de pruebas) y en uso (ejemplo en un entorno pre productivo) las cuales se encuentran documentadas en las normas ISO/IEC 25021, 25022, 25023 y 25024. La norma ISO/IEC 25040 define el proceso de

evaluación de la calidad del producto software, en el cual se deben establecer los requisitos, especificar, diseñar, ejecutar y concluir dicho proceso. Está familia se complementa con la recomendación ISO/IEC 25030 para la especificación de requisitos de calidad de productos de software en la etapa de elicitación o como entrada del proceso de evaluación.

La Integración de Sistemas de Modelos de Madurez de Capacidades (CMMI) se basa en que la calidad final de un producto depende directamente de la calidad de los procesos que los generan y mantienen. Bajo el enfoque de mejora continua, estos deberían ir incrementando su capacidad y, por consiguiente, aumentará el grado de madurez de la organización. Para lograr esto, el manual CMMI-DEV (CMMI, 2010) propone prácticas agrupadas en grupos conocidos como "áreas de procesos", ya que ejecutadas conjuntamente logran alcanzar diversos objetivos tanto específicos como genéricos de cada área, cuyas categorías son Gestión de proyectos, Gestión de procesos, Soporte e Ingeniería.

A su vez, CMMI cuenta con niveles que describen un camino evolutivo recomendado para que una organización pueda mejorar sus procesos de manera de usarlos para desarrollar y mantener sus productos y servicios.

El proceso de gestión de la calidad propuesto por el *NIST* (Ross, Mcevilley, y Oren, 2018) define objetivos, criterios, actividades y tareas que se deben seguir para garantizar la calidad final del producto. Esta norma establece un conjunto de actividades para evaluar y mantener el nivel de calidad deseado:

i. Analizar los resultados de las evaluaciones realizadas.
ii. Realizar revisiones periódicas para garantizar el cumplimiento de políticas, estándares y procedimientos.
iii. Monitorear las mejoras propuestas a los procesos, productos y servicios.
iv. Planificar acciones correctivas que se accionen cuando no se logren los objetivos.
v. Planificar acciones preventivas en caso de que exista riesgo de no cumplir con los parámetros establecidos.

4. Solución, Implementación y Modelo

Aquí se responderán los interrogantes planteados a lo largo de la tesis, integrando contenidos y, por último, se expondrá una recomendación para el Ciclo de Vida de *sistemas críticos* dentro de un Sistema de Gestión Integrado en el marco de la protección de las infraestructuras críticas.

4.1 Integración y Solución a interrogantes planteados

La sección 2.1 *Desafíos de ciberdefensa y ciberseguridad en el desarrollo,* evidencia que la multiplicidad de definiciones y recomendaciones requiere que se establezca un consenso regional e internacional. En particular, se debe trabajar a favor de establecer qué y cuáles son Infraestructuras Críticas considerando la importancia y criticidad en la prestación de los diferentes servicios. Además, es imprescindible que cada Estado incorpore al análisis las consecuencias (como por ejemplo el daño material, pérdidas económicas y sociales) que pudiera ocasionar la afectación y perturbación de los servicios considerados esenciales para la sociedad. Por poner un ejemplo, es posible afirmar que la afectación al servicio eléctrico podría afectar las comunicaciones, la salud, el transporte, entre otros[13].

Además, se dejó en evidencia que gran parte de las vulnerabilidades *pueden atribuirse a la calidad del desarrollo de software o malas configuraciones predeterminadas*. Frente a ello, integrar prácticas, estándares y procedimientos con atributos de seguridad a la gestión del ciclo de vida del Desarrollo resulta primordial para mitigar dichas vulnerabilidades.

Para realizar una adecuada gestión del ciclo de se deberá:

- Detectar las funciones que ejecutará el software y definir la criticidad de las mismas. Por cada función detectada elaborar un requerimiento de seguridad con su análisis correspondiente.
- Tener una consideración integral respecto de la seguridad del software.
- Todo lo detectado debe ser documentado y compartido a fin de retroalimentar el ciclo de vida y permitir la mejora continua.

Esto se debe a que una inadecuada especificación de requerimientos devendrá no solo en fallas en la seguridad sino en un incumplimiento del objetivo original del proyecto tal

[13] Un ejemplo de esto fueron los ciberataques que produjeron los cortes del servicio eléctrico en Brasil en noviembre del 2009 o los sufridos por Ucrania en diciembre del 2015 dejando a decenas de miles de personas sin servicios básicos.

como se describió en el punto 3.1.1. Como mínimo se deberán contemplar las siguientes fuentes de requerimientos:

- Las partes interesadas que deben expresar las preocupaciones y necesidades de seguridad.
- Implicaciones de seguridad de la especificación funcional.
- Requerimientos y normas de conformidad.
- Estándares de desarrollo seguro.
- Modelos de ataque y análisis de riesgo ambiental.
- Vulnerabilidades conocidas y probables en las tecnologías y componentes deban ser utilizados.

En el punto 3.1.2. expone la importancia de mantener por separando las entidades confiables de las entidades no confiables minimizando el número de puntos de entrada y salida dentro y fuera de cualquier entidad. También se comento la posibilidad de usar las pautas de planificación de calidad del TSP cuando no se cuenten con datos históricos de inyección de defectos.

El punto 3.1.3 remarca que los Estados deben ser precavidos al incorporar o consultar piezas de software ya que podrían estar contaminadas. Se deben realizar pruebas y evaluaciones en profundidad después de la adquisición o desarrollo y antes de la implementación o integración con las otras piezas o el sistema.

Previo al despliegue es recomendable realizar una evaluación de vulnerabilidades y una prueba de escaneo de superficie. Luego se debe realizar un endurecimiento ambiental agregando dispositivos externos como sistemas de detección de intrusos y firewalls tal como se expreso en el punto en el punto 3.1.4.

Para sistemas que contienen y manipulan datos críticos se debe considerar lo expuesto en la sección 3.2 *El concepto de Privacidad por Defecto,* donde se recomienda especificar el propósito para el cual se recopila, utiliza, mantiene y divulga cada pieza de información y establecer un límite de recolección para cada pieza de información o conjunto de datos. Para que esto sea sostenible en el tiempo, debe ser determinado en la etapa de definición de requerimientos del ciclo de vida aquí propuesto, además se garantizará la estrategia proactiva y preventiva del concepto de Privacidad por Defecto.

Por último, resulta imprescindible especificar la vida útil de la información y la forma de procesamiento de ésta en las primeras etapas del desarrollo del software estableciéndola

como configuración por defecto a fin de que la misma no pueda ser alterada una vez que el sistema se encuentre productivo.

Desde el punto de vista normativo y para subsanar los potenciales problemas que surjan del punto 3.2, el Estado deberá exigir a los desarrolladores, controladores y cualquier otra persona que se relacione con el ciclo de vida del que se tomen las medidas necesarias para garantizar la seguridad del producto. No solo debe existir la regulación sino que se debe auditar el cumplimiento de la misma ya sea para sistemas de operaciones, críticos, propios o contratados.

Si bien existen varios ejemplos de metodologías que aplican PbD, siendo algunas de las relevantes las expuestas en el punto 3.2.1, no se ha logrado generar un consenso sobre cual es la opción más abarcativa. Un posible acercamiento sería tener como base de la misma los siguientes puntos:

- Triada de la seguridad de información (confiabilidad, integridad, disponibilidad).
- Gestión del riesgo incorporando arboles de análisis de amenazas para detectar patrones.
- Tecnologías de mejora de la privacidad (PET) y control del ciclo de vida de la información.
- Tecnologías de capacidad de detección, relevación de información, disociabilidad y no repudio.

La etapa de desarrollo del CVS es el punto en donde se introducirán la mayoría de las vulnerabilidades, ya sea por una mala elicitación de requerimientos y/o deficiencias en el diseño las cuales se verán reflejadas en el código (teniendo este, también, errores introducidos por los programadores). Para mitigar este tipo de vulnerabilidades, se recomienda la aplicación del principio de corrección por construcción.

Para cumplir con este principio, la notación debe ser sólida y formal[14], debiendo estar dividida por etapas compatibles con herramientas para comprobar los prototipos que aquí se construyan tal como propone tanto CbC y el NIST. La información producida en la elicitación

[14] Esto quiere decir que la sintaxis del lenguaje se encuentra formalmente definida, teniendo su base semántica fundamentada en reglas matemáticas.

de requerimientos no debe repetirse en ninguna etapa, y se debe controlar que no se introduzca información redundante en las iteraciones.

El punto 3.3 demuestra que la necesidad de que la metodología sea incorporada en los desarrollos de sistemas vinculados al Estado o a infraestructuras críticas se justifica por el beneficio que se obtiene al evitar omisiones, errores o problemas en códigos que impliquen lógicas complejas.

La limitación planteada en ese punto debe ser superada mediante capacitaciones permanentes y la exposición de resultados a fin de lograr el cambio cultural necesario para obtener los beneficios del método de Corrección por Construcción.

Por último, las validaciones de las etapas manuales y/o automáticas expuestas en el punto 3.3 coinciden con lo expuesto en los puntos 3.4.1 en las recomendaciones existentes relacionadas a los conceptos de Seguridad por Defecto y por Diseño por el NIST y el CMMI en el punto 3.5.1.

Se propone la integración de lo puntos expuestos en la sección 3.5 en un único Sistema de gestión de calidad que contemple métricas internas, externas y de uso. Resulta imprescindible combinar herramientas de evaluación automática de calidad, las cuales deben ser eficientes, escalables y adaptables a distintos escenarios, con evaluaciones e interacciones humanas para garantizar un abordaje total del producto.

Las prácticas de calidad deben ser incorporadas al inicio del proyecto y sostenidas durante la duración de este bajo un enfoque de mejora continua, lo que aumentará la calidad del producto y la madurez del equipo de desarrollo siendo un valor agregado para futuros proyectos. Esto devendrá en que la calidad final de los productos desarrollados, al verse positivamente afectada por la calidad de los procesos que los generan y mantienen, sea la proyectada en el diseño de este y cumpla con todos los requisitos planteados en la primera etapa del ciclo de vida de éste.

4.2 Recomendación para la gestión segura del ciclo de vida de sistemas críticos

Se propone como definición de ciclo de vida al marco de referencia que contiene los procesos, actividades y tareas involucradas en el diseño, desarrollo, implementación y mantenimiento de una pieza de software que abarca toda la vida del sistema desde su diseño hasta su descarte.

El contexto para el desarrollo de software para sistemas críticos debe ser dentro de un Sistema de Gestión Integrado en donde el Ciclo de Vida adoptado será iterativo e

incremental, integrador de estándares, herramientas, métodos (manuales y automatizados) y sigue un plan de seguridad desde su inicio intentando dar respuesta al desafío planteado en el punto 3.1 e integrando lo expuesto en el punto 2.2.

Las fases de integración del Ciclo de Vida definidas en esta propuesta son:

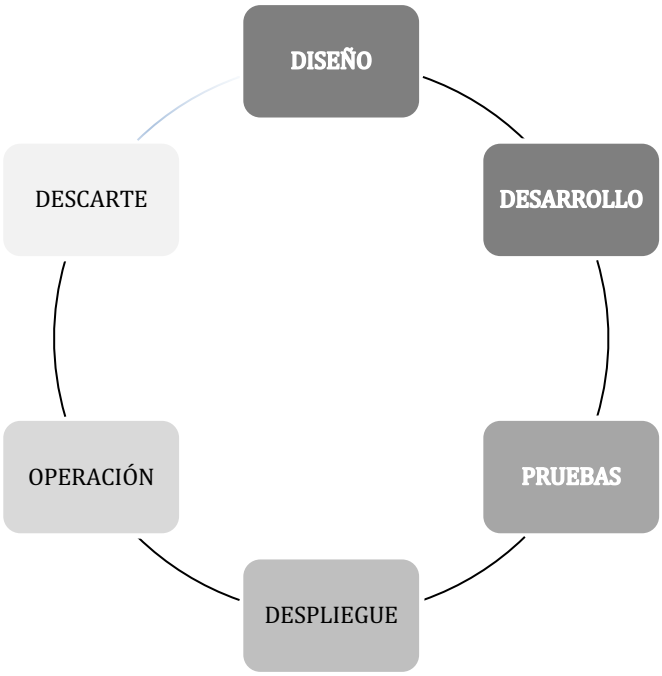


Figura 11. Fases de integración del Ciclo de Vida.

Cada una de estas fases debe ser documentada mediante el informe correspondiente que incluya, entre otros, los puntos de control de entrada y salida, cumplimiento de hitos, y su aprobación.

Como se expreso en el punto 3.1 cada una de estas fases tendrá un proceso de eliminación de defectos con el objetivo de detectar los mismos ni bien son introducidos en la pieza de software facilitando así la tarea de eliminación y reduciendo el impacto de estos en el proyecto mitigando cualquier vulnerabilidad producida.

4.2.1 Diseño.

La correcta ejecución de la fase inicial del Ciclo de Vida es crucial para el éxito del proyecto ya que allí se establecerán las bases y objetivos a cumplir. Las principales tareas de esta fase son:

- Definición de Requerimientos.
- Representación de los procesos mediante Diagramas de Flujo de Datos.

- Mapeo de las amenazas con los elementos del DFD agrupándolas en: Vincularidad; Capacidad de ser identificado; No repudio; Capacidad de Detección; Revelación de Información; Desconocimiento; Incumplimiento.
- Definición de aspectos de Seguridad de la Aplicación.
- Definición de la matriz de control de acceso a recursos.
- Definición de casos de uso, abuso y prueba.
- Diseño y modelización de Amenazas incluyendo el árbol de amenazas utilizando la clasificación STRIDE (3.1.2 Diseño).
- Diseño del protocolo de incorporación de nuevas piezas de software.
- Definición del protocolo de despliegue y mapeo de interconexiones de sistemas.

4.2.2 Desarrollo

La segunda fase se enfoca en llevar adelante lo definido en el Diseño. Aquí se deben utilizar técnicas de revisión de código, cumplir con estándares de programación definidos por la organización y realizar pruebas de control determinadas ya que el costo de detectar defectos es aún bajo (comparativamente con fases posteriores). Las principales tareas de esta fase son:

- Control de la arquitectura del entorno (firewalls, *cajas de arena*, control de firma, etc.)
- Revisión del código.
- Ejecución de pruebas de mutación de datos y privilegios mínimos.
- Control y eliminación de defectos.
- Control de los módulos de cifrado (utilizando por ejemplo la norma FIPS 140-2).

4.2.3 Pruebas

Es la fase especifica en donde se evaluará que todo lo realizado cumpla con los criterios establecidos en la fase de diseño. Las principales tareas son:

- Ejecución de herramientas de análisis estático y dinámico, analizadores de comportamiento ante mutación de datos, y otras pruebas definidas para analizar los parámetros de medición.
- Pruebas de respuesta a los modelos de amenazas.
- Pruebas de penetración.
- Pruebas de regresión de vulnerabilidades.

- Detección y eliminación de defectos.
- Pruebas de validez de datos de entrada a través de las interfaces de usuario y la transferencia a través de las interfaces de programación de aplicaciones (API) o entre redes en servicios y dispositivos.
- Corrección de todos los errores detectados.

4.2.4 Despliegue

Una vez que el desarrollo ha sido probado, se lo despliega en su entorno operativo con las interconexiones correspondientes cumpliendo con los procedimientos y pautas establecidas en la fase de Diseño. Las principales tareas de esta fase son:

- Detección y eliminación de defectos detectados en una evaluación de vulnerabilidades.
- Detección de interconexiones que produzcan fallas y vulnerabilidades.

4.2.5 Operación y Mantenimiento

Es la fase más extensa y una de las más costosas, por lo tanto, su correcto planeamiento y ejecución resultan cruciales para la vida del desarrollo, esto se debe a que el mismo debe mantenerse actualizado, operativo y con permanente control en sus interacciones a fin de garantizar que se cumpla con lo definido en la fase de diseño de forma constante. Las principales tareas de esta fase son:

- Entrenamiento de Seguridad a los usuarios de la Aplicación.
- Monitoreo.
- Pruebas de estado de la Aplicación.
- Prueba de los módulos de cifrado (utilizando por ejemplo la norma FIPS 140-2).
- Prueba de los protocolos de cifrado y comunicación (validando la recomendación AGA 12, entre otras).
- Controlar la ingeniería inversa que pueda hacerse sobre las aplicaciones a partir del almacenamiento de credenciales en la codificación.
- Generación de mecanismos de detección automática de cambios en el sistema notificando al personal responsable y cortando todas las comunicaciones en caso de que la pieza afectada sea crítica.
- Elaboración y generación de medidas de resiliencia para mantener el sistema disponible y operativo ante ataques de denegación de servicio.

- Recomendaciones de operatividad siguiendo la recomendación API Sec (entre otras).
- Actualización de sistemas relacionados y endurecimiento ambiental.
- Prueba periódica del plan de gestión y respuesta a incidentes.

4.2.6 Descarte

La última fase suele ser ignorada en la mayoría de los proyectos y, desde el punto de vista operativo, es una de las más importantes ya que se debe definir el destino de todo el volumen de datos generados durante la vida útil del sistema y la forma en que estos serán almacenados o migrados a futuras aplicaciones. Aquí también se debe definir el proceso de destrucción del sistema, sin que esto implique problemas de seguridad para su entorno. Las principales tareas de esta fase son:

- Elaboración del protocolo de eliminación de la aplicación y sistemas interconectados.
- Elaboración del protocolo de migración y resguardo de datos para futuras operaciones.

5. Conclusiones

Como se expresó inicialmente el *problema* que dio origen a la presente tesis se puede segmentar de la siguiente forma:

I. Existen múltiples organismos que emiten recomendaciones, normas y buenas prácticas relacionadas con la programación de sistemas y elementos informáticos.
II. Los Estados desarrollan sus propias prácticas, leyes y recomendaciones.

Estos dos puntos evidencian la heterogeneidad de criterios a la hora de decidir qué estrategia abordar para iniciar un proyecto de desarrollo de aplicaciones y sistemas críticos. Por lo tanto, la falta de estándares que sean un punto de encuentro entre los dos segmentos mencionados en lo relacionado con ciclo de vida del software, las técnicas de aseguramiento de la triada (confidencialidad, integridad y disponibilidad) y los atributos propios de los sistemas que soportan las infraestructuras críticas nacionales, trae como consecuencia que al tener que escoger una recomendación por sobre otras no se ponderen los riesgos que pueden impactar en la seguridad del proyecto y, por tratarse de sistemas e infraestructuras críticos y del Estado.

Para elaborar el presente trabajo se realizó un análisis de conceptos, normas, prácticas, recomendaciones y metodologías internacionalmente aceptadas para el desarrollo de sistemas y aplicaciones. En particular, se hizo hincapié en aquellos de carácter crítico utilizados para la generación, tratamiento, análisis y explotación de datos y análisis de vulnerabilidades desde un enfoque defensivo y disuasivo.

Para completar el análisis se realizó una extensa consulta de producciones académicas y científicas intentando abarcar las más representativas de ese universo (utilizando como parámetro las menciones y valoraciones dentro de cada organismo / lugar de consulta).

Como resultado de esta consulta, quedó en evidencia la multiplicidad de criterios relacionados al desarrollo de software y a los sistemas críticos que han ido evolucionando con el tiempo, al igual que la evolución en la influencia de la Ciberseguridad en la sociedad.

También se expuso y fundamentó la necesidad de integración de los precitados elementos, que se ve acrecentada por la interconexión de las infraestructuras críticas, sus sistemas y un mundo hipercomunicado cada vez más expuesto a *ciberamenazas*. Quedó demostrado que por más costosa o laboriosa que parezca la ejecución del proceso aportado en el punto 3.3, al realizar una evaluación del impacto económico que puede devenir de una vulnerabilidad en un sistema crítico se ahorrarán recursos financieros y se salvaguardará la

vida humana, un ejemplo de dichos costos se encuentran detallados en el punto 2.1.3 por el RTI.

El creciente uso de los dispositivos *IoT* y el trabajo remoto aumentan exponencialmente la superficie de ataque por lo cual es esperable que aumenten los ciberataques y la generación de *malwares* tanto de otros Estados como de organizaciones no gubernamentales. Frente a este escenario resulta prioritario mitigar las vulnerabilidades producidas o potenciadas por errores introducidos en el ciclo de vida, a fin de garantizar el control y la protección de la soberanía del ciberespacio nacional.

En el presente trabajo final de maestría se propone un sistema de gestión integrado con un ciclo de vida seguro que, lejos de repetir lo que hacen otros, reúne en un único sistema el proceso que podría emplearse para desarrollar aplicaciones y sistemas seguros que soportan servicios esenciales para la sociedad. Efectivamente, el aporte presentado es brindar un *sistema de gestión integrado* para la totalidad del *ciclo de vida* de desarrollo de software que se caracteriza por ser iterativo e incremental aplicable a cualquier tipo de sistema, y tiene su base en el estudio y análisis de las metodologías y normas que se emplean actualmente en el mundo.

A partir de lo planteado en el presente trabajo, queda pendiente para futuras investigaciones la elaboración de planes, procesos y procedimientos que den cumplimiento a las tareas de las fases de diseño, desarrollo, despliegue, operación y mantenimiento y descarte teniendo en cuenta un enfoque de ciberseguridad y con aplicación a los sistemas de las infraestructuras críticas. Estas son:

Fase de Diseño: Métodos de mapeo de amenazas con los elementos del DFD agrupándolas en: Vincularidad; Capacidad de ser identificado; No repudio; Capacidad de Detección; Revelación de Información; Desconocimiento; Incumplimiento.

Fase de Desarrollo: Herramientas adaptativas de Revisión de código.

Fase de Despliegue: Mecanismos de control de interconexiones a fin de detectar fallas o vulnerabilidades.

Fase de Operación y Mantenimiento: Mecanismos de detección de cambios en el sistema de forma automática, notificando al personal responsable y cortando todas las comunicaciones en caso de que la pieza afectada sea crítica.

Fase de Descarte: Aspectos relacionados a la migración de datos luego de la eliminación de las aplicaciones desarrolladas y sus sistemas interconectados.

Asimismo, quedará para posibles investigaciones futuras, la integración de lo analizado en una plataforma de Desarrollo que pueda ser presentado como propuesta de desarrollo de software seguro por parte del Estado Nacional, permitiendo incrementar la seguridad y los conceptos planteados por este proyecto.

A sabiendas la importancia que representan las infraestructuras críticas para cualquier Estado, el presente trabajo propone las recomendaciones que permitirán asegurar dichas infraestructuras desde el inicio del ciclo de vida de cualquier desarrollo maximizando los beneficios que conlleva, disminuyendo vulnerabilidades y por ende la probabilidad de ocurrencia de un eventual ciberataque que pudiera afectar los sistemas críticos, y accesoriamente resguarda los derechos de los ciudadanos, así como consolida la soberanía nacional.

Bibliografía

Agencia Española de Protección de Datos (AEPD). (2018). Guía práctica para las Evaluaciones de Impacto en la Protección de los datos sujetas al RGPD. Extraído el 25 de octubre del 2019 de: https://www.aepd.es/

Agile Methods and the Agile Manifesto. (n.d.). *Agile Software Construction*, 9-30. doi:10.1007/1-84628-262-4_2

Al-Nashif, Y. Baalbaki, B.A., Hariri, S. y Kelly, D. (2013). *Autonomic Critical Infrastructure Protection (ACIP) System*. IEEE. ISBN: 978-1-4799-0792-2/13.

Alcaraz, C. y Zeadally, S. (2015). *Critical Infrastructure Protection: Requirements and Challenges for the 21st Century*. International Journal of Critical Infrastructure Protection (IJCIP), vol. 8, Elsevier Science.

API. (2009). *API-1164: Pipeline SCADA Security*. American Petroleum Institute.

Arcilla, M. De La Cámara, M. Calvo, J. A. Sáenz, J. (2015). *Security by Design factors for developing and evaluating secure software. In: "2015 10th Iberian Conference on Information Systems and Technologies: CISTI 2015: Aveiro, Portugal, 17-20 June 2015"*, 17-20 de Junio de 2015, Aveiro, Portugal. ISBN 978-1-4799-8330-8. pp. 106-111. DOI: 10.1109/CISTI.2015.7170500.

Boehm, Penedo, Stuckle, Williams, y Pyster. (1984). *A Software Development Environment for Improving Productivity*. Computer, 17(6), 30-44. doi:10.1109/mc.1984.1659160

Caballero, I., García O. y Piattini G. (2007). *Calidad de sistemas informáticos.* México. Alfaomega Grupo Editor.

Capers, J. (2000). *Software Assessments, Benchmarks, and Best Practices*. Addison-Wesley.

CARI. (2013). *Ciberdefensa-Ciberseguridad: Riesgos y Amenazas*. Consejo Argentino para las Relaciones Internacionales. Extraído el 23 de Noviembre del 2019 de: http://www.cari.org.ar/pdf/ciberdefensa_riesgos_amenazas.pdf

Castellaro, M., Romaniz, S., Ramos, J. y Pessolani, P. (2016). *Hacia la Ingeniería de Software Seguro*. Facultad Regional Santa Fe - Universidad Tecnológica Nacional.

Cavoukian, A. (2014). Privacidad por diseño: de la retórica a la realidad. Comisionado de Información y Privacidad de Ontario.

Chapman, R. y Croxford, M. (2005). *Correctness by Construction: A Manifesto for High-Integrity Software*. The Journal of Defense Software Engineering.

Chapman, R. y Hall, A. (2002). *Correctness by Construction: Developing a Commercial Secure System*.IEEE Software Jan/Feb 2002, pp18-25.

Chapman, R. y Hall, A. (2004). *Correctness by Construction: Software Engineering.* Praxis Critical Systems.

Chapple, M. Gibson, D. y Stewart, J. (2015). *CISSP: certified information systems security professional study guide*. John Wiley & Sons, Indiana. USA. ISBN: 978-1-119-04271-6

Cleophas, L., Kourie, D. G. y Watson, B.W. (2015). *Experience with correctness by construction*. Science of Computer Programming, Vol. 97, Part 1. Elsevier. DOI: 10.1016/j.scico.2013.11.024

CMMI. (2010). *CMMI for Development, Version 1.3: Mejora de los procesos para el desarrollo de mejores productos y servicios*. CMU/SEI-2010-TR-033

Colesky, M., Hoepman, J. y Hillen, C. (2016). Un análisis crítico de las estrategias de diseño de privacidad. 2016 talleres de seguridad y privacidad IEEE (SPW). DOI: 10.1109 / spw.2016.23

Colon, M., Fedchak, E., Goertzel, K. M., McGibbon, T., McKinley, H., Oh, L., Vienneau, R. Winograd, T. (2007). *Software Security Assurance: State-of-the-Art Report (SOAR)*. Information Assurance Technology Analysis Center (IATAC) y Data and Analysis Center for Software (DACS)

Correa, G. J., Lacal-Arántegui, R., Yusta, J. M. (2011). *Methodologies and Applications for Critical Infrastructure Protection: State-of- the-Art*. Energy Policy, 39(10): 6100–6119. DOI: 10.1016/j.enpol.2011.07.010

Counter-Terrorism Committee Executive Directorate (CTED) & United Nations Office of Counter-Terrorism (UNOCT). (2018). The protection of critical infrastructure against terrorist attacks: Compendium of good practices. INTERPOL

CSRC. (2019). Computer Security Resource Center: Glossary. Consultado el 17 de Octubre del 2019 de https://csrc.nist.gov/Glossary

Cyber Security Agency of Singapore. (2016). *Singapore's Cybersecurity Strategy.* ISBN: 978-981110812-9

Cyber Security Agency of Singapore. (2017). *Security-by-Design Framework.* Extraído el 18 de septiembre del 2019 de: https://www.csa.gov.sg/legislation/supplementary-references

Davis, N. (2005). *Secure software development life cycle processes: A technology scouting report*. Carnegie Mellon University, Software Engineering Institute.

Decreto 577/2017. Boletín oficial de la República Argentina. Ciudad de Buenos Aires. 31 de Julio de 2017.

Decreto 703/2018 (*Derogado*). Boletín oficial de la República Argentina. Ciudad de Buenos Aires. 31 de Julio de 2018.

Deloitte. (2019). *DeVSecOps and the cyber imperative. Elevating, embedding, and evolving your risk response*. Deloitte Insights (Ed.), *Tech Trends 2019.* Beyond the digital frontier. (p. 102 - 116)

Department for Digital, Culture Media & Sport. (2018). *Secure by Design: Improving the cyber security of consumer Internet of Things Report*. Extraído el 10 de Octubre del 2019 de: https://www.gov.uk/government/publications/secure-by-design-report

Devanbu, P. T., y Stubblebine, S. (2000). *Software engineering for security. Proceedings of the Conference on The Future of Software Engineering - ICSE '00*. doi:10.1145/336512.336559

Diario Oficial de la Unión Europea (2016). Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respeta el tratamiento de datos personales y la libre circulación de estos datos y por el que se deroga la Directiva 95/46 / CE (Reglamento general de protección de datos) OJ L 119, 04.05.2016, p.1.

Disposición 6/2021. Boletín oficial de la República Argentina. Ciudad de Buenos Aires. 8 de Abril de 2021.

Donaldson, J. y Jenkins, J. (2000). *Systems Failures: An approach to understanding what can go wrong*. Middlesex University. ISBN 0-7695-0872-4

Dorneanu, B. Giannopoulos, G. Jonkeren, O. E. Ward, D. (2012). *Economic Impact Assessment of Critical Infrastructure Failure in the EU: A Combined Systems Engineering – Inoperability Input-Output Model*. Conference: The 20th International Input-Output Conference. Bratislava, Slovakia.

Fernando, O. N. N., Fernandopulle, W., Wijesiriwardana, C., & Wimalaratne, P. (2017). *Software Quality Assessment Tool: Source code evaluation tool for undergraduate and postgraduate projects*. 8th Annual International Conference on Computer Science Education: Innovation & Technology.

Ferrer, J., Hansen, M., Hoepman., Métayer, D., Tirtea, R., Schiffner, S. y Danezis, G. (2014). Privacidad y protección de datos por diseño, desde la política hasta la ingeniería. Heraklion: ENISA.

GAO. (2003). *Critical Infrastructure Protection: Challenges in Securing Control Systems*. United States General Accounting Office.

Goertzel, K. M. y Winograd, T. (2008). *Enhancing the Development Life Cycle to Produce Secure Software Version 2.0*. Defense Technical Information Center.

Hadley, M. y Huston, K. (2006). AGA 12, Part 2 Performance Test Plan. U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability.

Hansen, M. Jensen, M. y Rost, M. (2015). Protection Goals for Privacy Engineering. IEEE CS Security and Privacy Workshops.

Haralambiev, H., Boychev, S., Lilov, D. y Kraichev, K. (2011). *Applying source code analysis techniques. A case study for a large mission-critical software system*. IEEE. DOI: 10.1109/EUROCON.2011.5929241

Haridas, N. (2007). *Software Engineering – Security as a Process in the SDLC*. Information Security Reading Room. SANS Institute.

Hox, J. J., y Boeije, H. R. (2005). *Data Collection, Primary vs. Secondary*. Encyclopedia of Social Measurement, 593-599. doi:10.1016/b0-12-369398-5/00041-4

Humphrey, W. S. (2000). $The\ Team\ Software\ Process^{(SM)}(sm)$ $(TSP^{(SM)})$. Carnegie Mellon University, Software Engineering Institute.

IBM/Rational. (2001). *Rational Unified Process Best Practices for Software Development Teams*. Rational Software White Paper, no. TP026B Rev 11/012001

IEEE. (1990). *IEEE-STD 610.12-1990, IEEE standard glossary of software engineering terminology*.

IEEE. (1998). *IEEE Standard for a Software Quality Metrics Methodology*. IEEE. DOI: 10.1109/IEEESTD.1993.115124

IEEE. (2000). *P1402 Standard for physical security of electric power substations, IEEE 1402*.

IEEE. (2015). *IEEE Recommended Practice for Software Acquisition.* Software & Systems Engineering Standards (C/S2ESC) Committee of the IEEE Computer Society. ISBN: 978-1-5044-0085-5

IEEE. (2015). *IEEE Std* 90003 ™ − 2015. *IEEE Standard Adoption of ISO/IEC 90003:2014, Software Engineering— Guidelines for the Application of ISO 9001:2008 to Computer Software*. ISBN 978-0-7381-9850-7

INCIBE. (2015). *IEC 62443: Evolución de la ISA 99***.** Extraído el 27 de Octubre del 2019 de www.incibe-cert.es/blog/iec62443-evolucion-isa99

ISACA. (2016). Privacy Principles and Program Management Guide. Estados Unidos. ISBN: 978-1604206975

Jeun, I. y Zeballos A. G. (2016). *Best Practices for Critical Information Infrastructure Protection (CIIP): Experiences from Latin America and the Caribbean and Selected Countries*. Inter-American Development Bank and Korea Internet & Security Agency.

Jones, R. L., & Rastogi, A. (2004). Secure Coding: Building Security into the Software Development Life Cycle. *Information Systems Security, 13*(5), 29-39. doi:10.1201/1086/44797.13.5.20041101/84907.5

Karim, N. S., Albuolayan, A., Saba, T., y Rehman, A. (2016). *The practice of secure software development in SDLC: An investigation through existing model and a case study*. Security and Communication Networks, 9(18), 5333-5345. doi:10.1002/sec.1700

KU Leuven University. (2014) LINDDUN: a privacy threat analysis framework. KU Leuven, Leuven. Consultado el 5 de Octubre del 2019 en https://www.linddun.org/

Larman, C. (2004). *Agile & Iterative Development: A Manager's Guide*. Addison-Wesley. ISBN 9788177581591.

Lee, S. y Shon, T. (2016). *Open Source Intelligence Base Cyber Threat Inspection Framework for Critical Infrastructures*. ISBN 978-1-5090-4171-8/16

McGraw, G. Migues, S. y West, J. (2013). *Building Security In Maturity Model - BSIMM-V*. BSIMM.

Mesquida, A. L., Mas, A., San Feliu, T. y Arcilla, M. (2014). *Integración de Estándares de Gestión de TI mediante MIN-ITs*. Revista Ibérica de Sistemas y Tecnologías de la Información. DOI: 10.4304/risti.e1.31-45

Migues, S., Steven, J. y Ware, M. (2018). *Building Security In Maturity Model - BSIMM-10*. BSIMM.

Mosadeghi, R., Richards, R., y Tomlinson, R. (2017). *Chapter 8: Critical Infrastructure Protection and Uncertainty Analysis*, eds. Madu, C. y Kuei, C. *Handbook of Disaster Risk Reduction & Management*. DOI: 10.1142/10392

National Critical Information Infrastructure Protection Centre - NCIIPC. (2015). *Guidelines for the Protection of National Critical Information Infrastructure, Version 2.0*. New Delhi, India.

National Cyber Security Centre (NCSC). (2018). *Secure by Default.* Extraído el 15 de septiembre del 2019 de: https://www.ncsc.gov.uk/information/secure-default

Nevada González, P. A. (2014). *Propuesta de Modelo de certificación ISO/IEC 20000 combinando las mejores prácticas de ITIL V3 para el servicio de soporte a usuarios en empresas de servicios de IT. Caso de estudio: Empresa ITSTK*. Universidad de las Fuerzas Armadas ESPE. Ecuador.

Nickolov, E. (2005). *Critical Information Infrastructure Protection: Analysis, Evaluation and Expectations*. Information & Security. An International Journal, Vol.17, p. 105-119.

NIST. (2001). *Federal Information Processing Standards Publication: Security requirements for cryptographic modules (FIPS PUB 140-2)*. NIST.

NIST. (2013). *NIST Special Publication 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations.* DOI: 10.6028/NIST.SP.800-53r4

NIST. (2017). *NISTIR 8062: An Introduction to Privacy Engineering and Risk Management in Federal Systems*. DOI: 10.6028/NIST.IR.8062

NIST. (2019a). *Federal Information Processing Standards Publication: Security requirements for cryptographic modules (FIPS PUB 140-3)*. NIST.

NIST. (2019b). Vulnerability Type Change By Year - National Vulnerability Database. Extraído el 12 de Septiembre del 2019 de: https://nvd.nist.gov/general/visualizations/vulnerability -visualizations /cwe-over-time#vuln-type-change-by-year-title

Norma Internacional IEC 62351. (2018). *Series: Power systems management and associated information exchange - Data and communications security.*

Norma Internacional ISO 8402. (1994). Gestión de la calidad y aseguramiento de la calidad: Vocabulario.

Norma Internacional ISO 9000:2000. (2000). Sistemas de gestión de la calidad: Fundamentos y vocabulario.

Norma Internacional ISO 9001:2015. (2015). Sistemas de Gestión de la calidad

Norma Internacional ISO/IEC 17799 . (2005). *Information technology — Security techniques — Code of practice for information security management*.

Norma Internacional ISO/IEC 19791. (2015). *Information technology — Security techniques — Security assessment of operational systems*.

Norma Internacional ISO/IEC 25000 (2007). Evaluación de Calidad de Software.

Norma Internacional ISO/IEC 27000. (2018). *Information technology—Security techniques—Information security management systems—Overview and vocabulary*.

Norma Internacional ISO/IEC 27001. (2013). *Information technology—Security techniques—Information security management systems—Requirements*.

Norma Internacional ISO/IEC 27002. (2013). *Information technology—Security techniques—Code of practice for information security controls*.

Norma Internacional ISO/IEC/IEEE 24748-2:2018. (2018). *Systems and software engineering — Life cycle management*.

Norma Internacional ISO/IEC 29101:2013 (2018). *Information technology — Security techniques — Privacy architecture framework.*

OECD. (2008). *Malicious Software (Malware): A Security Threat to the Internet Economy*. Ministerial Background Report, DSTI/ICCP/REG(2007)5/FINAL. Paris.

OWASP. (2016). *Security by Design Principles.* Extraído el 11 de septiembre del 2019 de: https://www.owasp.org/index.php/Security_by_Design_Principles

OWASP SAMM Project. (n.d.). Extraido el 10 de octubre del 2019 de https://www.owasp.org/index.php/OWASP_SAMM_Project

Persano, M.L. (2015). *El desarrollo de software en Sistemas Críticos para la Seguridad (Tesis de grado)*. Instituto Universitario Aeronáutico. Buenos Aires.

President's Information Technology Advisory Committee - PITAC. (1999). *Report to the President - Information Technology Research: Investing in Our Future*. USA

Radatz, J., Olson, M. y Campbell, S. (1995). *CrossTalk: The Journal of Defense Software Engineering*. MIL-STD-498. USA

Rehak, D. Senovsky, P. y Slivkova, S. (2018). *Resilience of Critical Infrastructure Elements and Its Main Factors*. Technical University of Ostrava, Czech Republic;

Resolución 1523/2019. Ciudad de Buenos Aires, Argentina, 12 de septiembre del 2019.

Ross, R., Mcevilley, M., y Oren, J. C. (2018). *Systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems*. NIST Special Publication 800-160 volume 1. DOI:10.6028/nist.sp.800-160v1

Royce, W. (1970). *Managing the Development of Large Software Systems, Concepts and Techniques*. Proceedings of IEEE. USA

RTI. (2002). *Planning Report 02-3: The economic impacts of inadequate infrastructure for software testing*. NIST: Program Office Strategic Planning and Economic Analysis Group.

Shirazi, H. M. (2009). *A New Model for Secure Software Development*. International Journal of Intelligent Information Technology Application p. 136-143

Schlegel, R., Obermeier, S., & Schneider, J. (2015). *Assessing the Security of IEC 62351*. DOI: 10.14236/ewic/ICS2015.2

Sieńko, P. (2015). *Methods of securing and controlling critical infrastructure assets allocated in information and communications technology sector companies in leading*. Securitologia, vol. 22, no. 2, pp. 107–123.

Smart Grid Interoperability Panel. (2010). *Introduction to NISTIR 7628: Guidelines for Smart Grid Cyber Security*. Cyber Security Working Group.

Software Assurance Forum for Excellence in Code (SAFECode). (2018). *Fundamental Practices for Secure Software Development: Essential Elements of a Secure Development Lifecycle Program*. 3ª ed. SAFECode.

Steidl, D., Deissenboeck, F., Poehlmann, M., Heinke, R. y Uhink-Mergenthaler, B. (2014). *Continuous Software Quality Control in Practice*. IEEE. DOI: 10.1109/ICSME.2014.95

Unión Internacional de Telecomunicaciones - ITU (2008). *Recomendación X.1205 (04/08): Aspectos generales de la ciberseguridad*.

US Dept. of Energy. (2005). *National SCADA Test Bed: A Summary of Control System Security Standards Activities in the Energy Sector*. U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability.

US Dept. of Homeland Security and National Institute of Standards and Technology. (2011). *Modeling and Simulation of Critical Infrastructure Systems for Homeland Security Applications*. Workshop on Homeland Security Modeling & Simulation.

US Dept. of Homeland Security and National Institute of Standards and Technology. (2013). *National Infrastructure Protection Plan: Partnering for Critical Infrastructure Security and Resilience*.

US. Dept. of Homeland Security. (2013). NIPP 2013: Partnering for Critical Infrastructure Security and Resilience. US. Department of Homeland Security.

US Dept. of Homeland Security and National Institute of Standards and Technology. (2015). *US Presidential Policy Directive PPD21, Critical Infrastructure Security and Resilience*.

US Dept. of Homeland Security. (2011). *Catalog of control systems security: Recommendations for standards developers*.

US Government Accountability Office. (2017). *Critical Infrastructure Protection: DHS Risk Assessments Inform Owner and Operator Protection Efforts and Departmental Strategic Planning*. GAO-18-62

Wuyts, K. (2015). "Doctoral thesis: Privacy Threats in Software Architectures", Arenberg Doctoral School, KU Leuven, Leuven.

Anexos

Anexo 1. Directrices de Calidad del TSP

Medida	Objetivo	Comentario
Porcentaje libre de Defectos		
Compilar	> 10%	
Pruebas unitarias	> 50%	
Pruebas de integración	> 70%	
Pruebas de Sistema	> 90%	
Defectos / KLOC		
Total de defectos inyectados	75 - 150	En caso de no estar entrenado en PSP utilizar 100 – 200
Compilar	< 10	Todos los defectos
Pruebas unitarias	< 5	Todos defectos importantes (en LOC)
Pruebas de integración	< 0.5	Todos defectos importantes (en LOC)
Pruebas de Sistema	< 0.2	Todos defectos importantes (en LOC)
Defectos relacionados		
Defectos de revisión de diseño detallado / defectos de pruebas unitarias	> 0.2	Todos defectos importantes (en LOC)
Defectos de revisión de código / defectos de compilación	> 0.2	Todos defectos importantes (en LOC)
Defectos relacionados al Desarrollo		
Revisión de requerimientos / tiempo para requerimientos	> 0.25	Elicitación en el tiempo requerido
Inspección del diseño de alto nivel / tiempo para diseño de alto nivel	> 0.5	Solo trabajo de diseño, no estudios
Diseño detallado / tiempo para codificación	> 1.00	
Revisión de diseño detallado / tiempo para el diseño detallado	> 0.5	
Revisión de código / tiempo para codificación	> 0.5	
Tasas de revisión e inspección		
Páginas de requisitos / hora	< 2	Páginas de texto a espacio simple
Páginas de diseño de alto nivel / hora	< 5	Lógica de diseño formateada
Diseño detallado de líneas de texto / hora	< 100	Pseudocódigo ~ igual a 3 LOC
Código LOC / hora	< 200	LOC lógico
Tasas de inyección y eliminación de defectos		
Defectos inyectados en requerimientos / hora	0.25	Solo defectos mayores
Defectos eliminados en la inspección de requerimientos / hora	0.5	Solo defectos mayores
Defectos inyectados en el diseño de alto nivel / hora	0.25	Solo defectos mayores
Defectos eliminados en la inspección de diseño de alto nivel / hora	0.5	Solo defectos mayores
Defectos inyectados en el diseño detallado de / hora	0.75	Solo defectos de diseño
Defectos eliminados en la revisión detallada del diseño / hora	1.5	Solo defectos de diseño

Defectos eliminados en la inspección detallada del diseño / hora	0.5	Solo defectos de diseño
Defectos inyectados en la codificación / hora	2.0	Todos los defectos
Defectos eliminados en la revisión de código / hora	4.0	Todos defectos en la fuente de LOC
Defectos inyectados en la compilación / hora	0.3	Cualquier defecto
Defectos eliminados en la inspección de código / hora	1.0	Todos defectos en la fuente de LOC
Defectos inyectados en las pruebas / hora	0.067	Cualquier defecto
Rendimiento por Fase		
Inspección de requerimientos por equipos	~ 70%	Sin contar comentarios editoriales
Inspecciones y revisiones de diseño	~ 70%	Usando análisis de estado y tablas de rastreo.
Inspecciones y revisiones del código	~ 70%	Usar listas de verificación personales.
Compilación	~ 50%	Más del 90% de defectos de sintaxis.
Pruebas unitarias a ~ 5 o menos defectos/KLOC	~ 90%	Para defectos mayores / KLOC - 50-75%
Pruebas de integración y sistema a ~ < 1.0 defectos/KLOC	~ 80%	Para defectos mayores / KLOC - 30-65%
Antes de compilar	~ 75%	Asumiendo métodos de diseño de sonido.
Antes de las pruebas unitarias	~ 85%	Asumiendo verificaciones lógicas en las revisiones.
Antes de las pruebas de integración	~ 97.5%	Para productos pequeños, 1 defecto máx.
Antes de las pruebas del sistema	~ 99%	Para productos pequeños, 1 defecto máx.

Directrices de Calidad del TSP (Humphrey, 2000, p. 20)

Anexo 2. Tablas comparativas entre normas, estándares y recomendaciones

Se anexan tablas comparativas entre normas, estándares y recomendaciones recopilando el trabajo realizado por Alcaraz y Zeadally.

- Para sistemas de información y de comunicación SCADA: NIST 800-53 y NISTIR 7628.
- Recomendaciones generales: ISA 99-1, ISA 99-2, ISO 177799, ISO 27001, ISO 27002 y ISO 19791.

Subcontroles de seguridad operacional.	Estándares de organización y operaciones							
	NIST 800-53	NISTIR 7628	ISA 99-1	ISA 99-2	ISO 17799	ISO 27001	ISO 27002	ISO 19791
Adquisición de sistemas y servicios	X	X	X	X	X	X		X
Administración de las configuraciones	X	X	X	X	X	X		X
Protección de sistemas y comunicaciones	X	X	X	X	X	X	X	X
Administración de la información y los documentos	X	X		X	X	X		X
Desarrollo y mantenimiento de los sistemas	X	X	X	X	X	X	X	X
Administración y respuesta a incidentes	X	X		X	X	X	X	X
Integridad de la información y los sistemas	X	X			X	X		X
Control de acceso	X	X		X	X	X	X	X
Contabilidad y auditoría	X	X		X	X	X	X	X
Protección de medios	X	X			X	X		

Cumplimiento de estándares organizacionales y operacionales para sistemas de control crítico (Alcaraz, C. y Zeadally, S. 2015, P. 17)

- Servicios criptográficos: IEC-62351 y FIPS 140-2.
- Tecnologías de transporte de información: WirelessHART, ISA100.11 y ZigBee.

Sub controles de seguridad operacional	Estándares técnicos				
	IEC 62351	FIPS 140-2	Wireless HART	ISA100.11a	ZigBee
Adquisición de sistemas y servicios		X	X	X	X
Administración de las configuraciones		X	X	X	X

Protección de sistemas y comunicaciones	X	X	X	X	X
Administración de la información y los documentos		X	X	X	X
Desarrollo y mantenimiento de los sistemas	X	X	X	X	X
Administración y respuesta a incidentes	X	X	X	X	X
Integridad de la información y los sistemas	X	X		X	X
Control de acceso	X	X	X	X	X
Contabilidad y auditoría		X	X	X	X
Protección de medios					

Cumplimiento de normas técnicas para sistemas de control crítico. (Alcaraz, C. y Zeadally, S. 2015, P. 19)

- Servicios criptográficos: AGA 12-1 y AGA 12-2.
- Marcos de protección: NERC CIP, GAO-04-140T, IEEE 1402 y API Sec.

Sub controles de seguridad operacional.	Recomendaciones y Guías					
	AGA 12-1	AGA 12-2	NERC CIP	GAO-04-140T	IEEE 1402	API Sec
Adquisición de sistemas y servicios	X		X	X		X
Administración de las configuraciones	X		X			X
Protección de sistemas y comunicaciones	X	X	X	X		X
Administración de la información y los documentos	X		X			X
Desarrollo y mantenimiento de los sistemas	X		X	X	X	X
Administración y respuesta a incidentes	X		X		X	X
Integridad de la información y los sistemas	X	X	X			X
Control de acceso	X	X	X	X	X	X
Contabilidad y auditoría	X		X			X
Protección de medios	X		X			X

Cumplimiento de normas técnicas para sistemas de control crítico. (Alcaraz, C. y Zeadally, S. 2015, P. 19)