# GUÍA DE BUENAS PRÁCTICAS DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE CON BASE EN ESTÁNDARES RECONOCIDOS EN EMPRESAS DE DESARROLLO DE SOFTWARE

CAMILO FERNANDEZ BERNAL
Ingeniero de Sistemas y Computación

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA (ECBTI)
ESPECIALIZACION EN SEGURIDAD INFORMATICA
PEREIRA - RISARALDA
2018

# GUÍA DE BUENAS PRÁCTICAS DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE CON BASE EN ESTÁNDARES RECONOCIDOS EN EMPRESAS DE DESARROLLO DE SOFTWARE

CAMILO FERNANDEZ BERNAL
Ingeniero de Sistemas y Computación

Trabajo de grado para optar al título de especialista en Seguridad informática

Director
ANÍVAR CHAVES TORRES
Ingeniero de sistemas

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA (ECBTI)
ESPECIALIZACION EN SEGURIDAD INFORMATICA
PEREIRA - RISARALDA
2018

Nota de Aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Pereira, Risaralda. Septiembre de 2018

EXCLUSIÓN DE RESPONSABILIDAD

Yo, Camilo Fernández Bernal, identificado con cedula de ciudadanía No. 1’088.303.409 de la ciudad de Pereira, Risaralda, actuando en nombre propio y en calidad de autor del presente trabajo de grado manifiesto que la información consignada en este documento es una producción original y no infringe los derechos de autor de ningún tercero, y no compromete la ideología de la Universidad Nacional Abierta y a Distancia.

A mi único y principal motor en mi vida, mi familia, que sin ellos no podría ser quien soy ahora y a quienes les debo todo.

A mis profesores quienes de una u otra manera han sido parte de mi proceso de formación y contribuido a mi desarrollo académico y profesional.

**CONTENIDO**

LISTA DE CUADROS

Pag.

## DICCIONARIO DE TERMINOS

Durante el desarrollo del proyecto se utilizaron diferentes términos técnicos que son necesarios precisar para un mayor entendimiento:

Amenaza: Una amenaza es una violación potencial de la seguridad. No es necesario que la violación ocurra para que la amenaza exista.

Calidad: Es una propiedad inherente de una cosa que permite categorizarla y medirla frente a las demás.

CMMI: Fue diseñado como un modelo de mejora de la capacidad que fácilmente se puede adaptar para resolver cualquier inconveniente por un rendimiento irregular en cualquier nivel o área de la organización en cualquier industria. El modelo proporciona directrices y recomendaciones para ayudar a su organización a diagnosticar problemas y mejorar el rendimiento.

Estandarización: Garantiza que los procesos de una organización sean realizados de forma uniforme por parte todos los involucrados en el.

ISO 27001: Es la norma principal de la serie 27000 ya que reúne el conjunto de requisitos y controles necesarios para implementar un sistema de seguridad de la información; permite identificar y evaluar los riesgos físicos y lógicos para así definir políticas y estrategias que permitan reducir la incidencia de estos riesgos y poder salvaguardar la información. Tiene un enfoque basado en procesos que hace uso del ciclo de mejora continua PHVA (Planificar, Hacer, Verificar y Actuar).

Metodología: Conjunto de métodos, procedimientos y técnicas definidos para interpretar o resolver diferentes problemas.

PSP: El Personal Software Process (PSP) proporciona a los ingenieros un marco disciplina personal para hacer el trabajo de software. El proceso de PSP consiste en un conjunto de métodos, formas y secuencias de comandos que muestran a los ingenieros de software cómo planificar, medir y gestionar su trabajo. PSP está diseñado para su uso con cualquier lenguaje de programación o metodología de diseño y se puede utilizar para la mayoría de los aspectos del trabajo de software, incluidos los requisitos de escritura, ejecución de pruebas, los procesos de definición, y la reparación de los defectos.

Seguridad de la Información: se basa principalmente en la protección de los datos de una organización independiente del medio en el que se encuentre; lo que busca en si es que la información este siempre disponible, se conserve tal cual como fue concebida y solo sea accesible por el personal autorizado para este fin.

TSP: Guía a los equipos de ingenieros que están desarrollando productos intensivos en software. El uso de TSP ayuda a las organizaciones a establecer una práctica de ingeniería madura y disciplinada que produce software seguro y fiable en menos tiempo y con menores costes.

Vulnerabilidad: Nivel de exposición o deficiencia de un sistema ante un posible riesgo de seguridad.

# INTRODUCCION

El creciente uso de las tecnologías de la información y la comunicación ha denotado también un incremento exponencial del número de amenazas existentes en la red, lo que ha generado la necesidad de desarrollar sistemas de información y aplicaciones más seguras que garanticen la integridad, confidencialidad y disponibilidad de los datos.

En la actualidad la gran mayoría de empresas han optado por caracterizar y sistematizar toda su información almacenándola en sistemas que faciliten su administración y están acordes con las exigencias del mercado. No es para menos la preocupación de las organizaciones por salvaguardar sus datos, ya que hoy en día la información se ha constituido como el activo más importante que poseen y ante tantas amenazas es necesario protegerse de la mejor manera.

Para las empresas de desarrollo software, al igual que para desarrolladores independientes, la seguridad es en una herramienta que mejora la calidad de sus productos y los protege ante cualquier vulnerabilidad que se pudiese presentar. Por consiguiente, ya no basta con desarrollar software por desarrollar, es necesario que se haga uso de buenas prácticas y metodologías de seguridad en el desarrollo que faciliten la creación de sistemas confiables que estén acordes con las exigencias del mercado y que incluyan el componente de seguridad en cada fase del proceso de implementación.

Para el desarrollo del presente estudio, en primera instancia se realizó la investigación y recopilación de las mejores prácticas de seguridad propuestas por estos estándares, posteriormente se clasificaron y midieron estas prácticas, paso seguido se identificaron los factores de riesgo asociados a la implementación de

estas prácticas seleccionadas, consolidando el desarrollo de este trabajo de investigación en un documento guía.

Como resultado se obtuvo una guía de buenas prácticas de seguridad para el desarrollo de software con base en una metodología de desarrollo seguro como lo es PSP/TSP, enmarcado bajo estándares internacionales de seguridad como lo sugiere la norma ISO 27001 y con la estandarización de los procesos descritos por las prácticas que implementa el nivel 2 de madurez de CMMI en las empresas de desarrollo de software.

# 1 EL PROBLEMA

## 1.1 DESCRIPCIÓN DEL PROBLEMA

El presente de las empresas de desarrollo de software y de los desarrolladores independientes está regido en gran medida por la ausencia del uso de metodologías y buenas prácticas de seguridad que faciliten el proceso de fabricación de nuevos sistemas de información y aplicaciones, la gran mayoría opta por aplicar parcialmente una de las tantas metodologías de desarrollo existentes, que muchas veces no logran estandarizar este proceso y provoca que siempre que se quiere iniciar un nuevo proyecto el proceso de elaboración este basado en la improvisación.

Esta falta de estandarización y uso de buenas prácticas de seguridad, aparte de generar posibles riesgos de aparición de alguna vulnerabilidad en el producto desarrollado, implica grandes pérdidas de recursos porque se tendrían que realizar reprocesos para la revisión y ajuste de las fallas de seguridad que pueda tener el sistema, además de los tiempos y personal que se deberán dedicar en su reparación, la pérdida de confianza del cliente, sin mencionar el hecho de que si el software desarrollado entra en producción y alguna de sus vulnerabilidades es explotada por un tercero puede generar aún mayores pérdidas para la organización.

Otro escenario a tener en cuenta es la identificación y análisis de riesgos a través del ciclo de vida del software seguro como lo menciona Gloria Piedad Gasca[1], ya que las amenazas están presentes en cada fase del ciclo y pueden representar un riesgo si no son identificadas de forma oportuna; por tanto, es necesario clasificar,

---

[1]GASCA HURTADO, Gloria P. Análisis de riesgos para el desarrollo de software seguro. Universidad Politécnica de Madrid [En línea]. Agosto de 2006. [citado 29 junio de 2018]. Disponible en: <http://www.dlsiis.fi.upm.es/docto_lsiis/Trabajos20052006/Gasca.pdf>

cuantificar el impacto y dar un tratamiento adecuado a los riesgos en la medida de que se apliquen buenas prácticas para el desarrollo durante todo el proceso de implementación.

Estos problemas sugieren la falta de herramientas estandarizadas que faciliten el desarrollo seguro de software con un alto grado de calidad, en donde el proceso de desarrollo sea igual para todos los proyectos, con unos parámetros y actividades bien definidas que faciliten el proceso de implementación y mitigación de riesgos, es por eso que surge la necesidad de una guía que contemple un conjunto de buenas prácticas para el desarrollo, ágil, seguro y estandarizado de software, enmarcado en los controles y normas internacionales.

## 1.2 FORMULACIÓN DEL PROBLEMA

¿Cómo facilitar la mitigación de amenazas, vulnerabilidades y riesgos durante las diferentes fases del desarrollo de software con base en estándares reconocidos por empresas de desarrollo de software?

## 1.3 OBJETIVOS

### 1.3.1 Objetivo General

Diseñar una guía de buenas prácticas de seguridad en el desarrollo de software con bases en estándares reconocidos en empresas de desarrollo de software.

### 1.3.2 Objetivos Específicos

- Investigar y recopilar la información de las buenas prácticas de las metodologías de seguridad escogidas (PSP/TSP, CMMI Nivel 2 y la norma ISO 27001) que están relacionas con seguridad en el desarrollo de software

- Categorizar las buenas prácticas de las metodologías de seguridad escogidas para el desarrollo de software.

- Identificar los factores de riesgo asociadas a las prácticas seleccionadas.

- Sistematizar la información sobre las buenas prácticas de seguridad en el desarrollo de software y documentarlas para que sean fácilmente comprensibles y aplicables por los desarrolladores.

## 1.4 JUSTIFICACIÓN

Si los procesos de desarrollo de software no se ejecutan eficientemente garantizando la seguridad y calidad de los mismos podría generar grandes riesgos para la información almacenada en esos sistemas desarrollados, el hecho de contemplar los controles de la norma ISO 27001 que rigen sobre el desarrollo seguro y el acceso a los sistemas de información facilitaría en gran medida la protección de los datos, además de dar cumplimiento a estándares internacionales en seguridad de tecnologías de la información.

Garantizar la seguridad en el desarrollo de software no solo se basa en el uso y aplicación de una metodologías, normas, disposiciones legales o conjunto de prácticas durante el desarrollo, sino que se puede complementar a través de la definición formal de un proceso en donde se estandarice la forma en que se debe realizar la implementación del software con miras a disminuir el error mediante la optimización y calidad del producto como lo describe algunas de las practicas descritas por CMMI en su nivel dos de madurez.

Por consiguiente, la elaboración de una guía de buenas prácticas de desarrollo de software que contenga las actividades más destacadas descritas por una metodología segura de desarrollo como PSP/TSP, bajo un conjunto de prácticas para la estandarización del ciclo de vida del software como lo expone CCMI nivel dos de madurez, todo esto enmarcado en los controles establecidos por la norma ISO 27001 concernientes al desarrollo de software facilitarían el proceso de implementación de nuevas soluciones de software optimizando el recurso, garantizando la calidad y resguardando efectivamente la información.

La implementación de una guía de buenas prácticas de seguridad, basada en estándares reconocidos, en el ciclo de vida de desarrollo de software mejoraría la

confiabilidad y calidad del producto generado, crearía una experiencia satisfactoria para el desarrollador, en donde mediante el uso de metodologías y prácticas bien definidas podrá usar eficientemente los recursos a su disposición, se mitigaría considerablemente los riesgo presentes durante todas las fases del desarrollo, también representaría un mayor grado de confianza y satisfacción de parte del cliente en el producto generado y por consiguiente en la empresa.

# 2 MARCO DE REFERENCIA

## 2.1 ANTECEDENTES

El concepto de seguridad informática ha evolucionado progresivamente a través del tiempo en proporción al crecimiento y complejidad del volumen de información, para lo cual diferentes autores han desarrollado numerosas y diversas practicas y metodologías en las que describen acciones que permiten de una u otra manera salvaguardar los datos, reducir los riegos y mejorar la calidad de los procesos de seguridad implementados. A continuación, se presentan los antecedentes de la investigación, sobre el tema desarrollado en el presente proyecto, de tal forma que permita una mayor comprensión e interpretación sobre el uso de metodologías y buenas practicas para la seguridad de la información.

Daniel Port, Rick Kazman y Ann Takenaka [2] en su artículo "Strategic Planning for Information Security and Assurance" plantean una serie de estrategias para proporcionar un nivel razonable de seguridad por un costo razonable de recursos proporcional al riesgo tratado, en las cuales dependiendo de las salvaguardas y mecanismos de control utilizados para la mitigación se puede lograr la eliminación, aceptación, control o transferencia de riesgos dentro de lo que es realmente factible bajo las condiciones y limitaciones particulares. Para la elección de la estrategia más óptima en reducción del riesgo definen un algoritmo que tiene en cuenta los principales atributos de riesgo existentes, los posibles tratamientos o salvaguardas para cada atributo evaluados individualmente para el riesgo específico, además del costo, el tamaño y la probabilidad de pérdida de la misma, para finalmente calcular la matriz de costo para determinar cuál podría ser mejor. Esta investigación es

[2] PORT, Daniel; KAZMAN, Rick; TAKENAKA, Ann. Strategic Planning for Information Security and Assurance [En linea]. Department of Information Technology Management, University of Hawaii. IEEE. 2008. pp. 466-471.

acorde con los objetivos del proyecto ya que sirve de guía para determinar cuáles prácticas son más beneficiosas en materia de seguridad para el desarrollo.

D.P. Gilliam[3] investigador del Laboratorio de Propulsión a Chorro de California en su investigación “Security risks: management and mitigation in the software life cycle” explica que es necesario considerar los riesgos implícitos en cada fase del ciclo de desarrollo de software y especificar los requerimientos de seguridad, que deben abordar las necesidades identificadas en función del entorno y las necesidades del cliente, desde la concepción misma del proyecto, explica que el modelado y las pruebas basadas en propiedades son herramientas que pueden ayudar a reducir los riesgos de seguridad ya que permiten evaluar el código en contraste con los requerimientos definidos, además junto con un buen instrumento de gestión de riesgos pueden disminuir las vulnerabilidades de seguridad en el ciclo de vida del software.

Manju Khari[4] de La India en su investigación “Embedding Security in Software Development Life Cycle (SDLC)” expresa como en el ciclo de desarrollo de software tradicional debe estructurarse un alto nivel de seguridad, esto introduciendo el triángulo de la CIA (Confidencialidad, integridad y disponibilidad) y con un componente de administración del riesgo.

Wenjin Wang[5] de china en su investigación “Development of Mass Spectrometer Software Project Based on CMMI” habla de cómo la calidad del software se ha convertido en el problema de cuello de botella en el rendimiento del espectrómetro de masas, impacto directo en la confiabilidad y mantenibilidad del producto, y como

---

[3] GILLIAM D.P.. Security risks: management and mitigation in the software life cycle [En linea]. Jet Propulsion Laboratory. California Inst. of Technology. Pasadena, California. IEEE. 2005. pp. 319 – 325.

[4] KHARI, Manju. Embedding Security in Software Development Life Cycle (SDLC) [En linea]. Dept. of Computer Science IEEE. 2016. pp. 2182 – 2186.

[5] WANG, Wenjin. Development of Mass Spectrometer Software Project Based on CMMI [En Linea]. Purification Equipment Research Institute of CSIC. IEEE. 2017. pp. 2508 – 2511.

el implementar el modelo CMMI se convierte en una elección para cambiar el modo de trabajo del desarrollo de software, para lograr una buena administración, para buscar un alto nivel calidad del software y para buscar un desarrollo seguro y eficiente.

Jim Whitmore y Will Tobin[6] proponen en "Improving Attention to Security in Software Design with Analytics and Cognitive Techniques" que las organizaciones de tecnologías de la información deben determinar cómo se organizará el desarrollo de un software dentro del ciclo de vida del software, es decir, la selección formal o informal, y cómo se manejan la seguridad y el riesgo dentro de los proyectos para activos de software específicos. Los autores afirman cuatro áreas de acción para la seguridad en el ciclo de vida del software que son: primero diseñar y construir la seguridad en el software desde adentro; segundo completar las actividades de aseguramiento antes del despliegue operacional; tercero responder a las amenazas y vulnerabilidades en los sistemas operativos y cuarto aplicar las lecciones aprendidas operacionalmente y como experiencias para notificaciones a futuras iteraciones del proceso de desarrollo.

I. Garcia e I. Andrea[7] en su investigación "Using the Software Process Improvement approach for Defining a Methodology for Embedded Systems Development using the CMMI-DEV v1.2" dicen que las decisiones de ingeniería de sistemas están siendo impulsadas por restricciones de hardware, que luego impactan los esfuerzos en el software dos etapas más adelante en el ciclo de vida cuando se desarrollan los requisitos de software a nivel de componente, por tanto sugieren optimizar la puntualidad, la productividad y la calidad del desarrollo de sistemas embebidos, sugieren que las empresas deben adaptar tecnologías de ingeniería de software

---

[6] WHITMORE, Jim; TOBIN, Will. Improving Attention to Security in Software Design with Analytics and Cognitive Techniques [En linea]. USA. IEEE. 2017. pp. 16 – 21.

[7] GARCIA, I; ANDREA, I. Using the Software Process Improvement approach for Defining a Methodology for Embedded Systems Development using the CMMI-DEV v1.2 [En linea]. Technological University of the Mixtec. Mexico. IEEE. 2010. pp. 233– 240.

apropiadas para situaciones específicas, por tal motivo realizaron la creación de la metodología SPIES basada en las áreas de procesos y practicas específicas de CMMI-DEV con un enfoque iterativo que garantiza que el proceso de desarrollo se verifique en cada fase, además los autores hacen uso de la idea básica de TSP de definir un conjunto de documentos guía que ayude a los desarrolladores en la gestión de los proyectos.

Yasemin Acar, Christian Stransky, Dominik Wermke, Charles Weir, Michelle L. Mazurek y Sascha Fahl[8] en su articulo de investigación "Developers Need Support, Too: A Survey of Security Advice for Software Developers" argumentan el hecho de que cada vez más los desarrolladores toman conciencia sobre la seguridad en el software y la necesidad de tener un código seguro, aunque manifiestan que si bien la red es el principal medio de consulta, por parte de los desarrolladores, sobre prácticas seguras muchas de estas no han sido verificadas y están cayendo de forma persistente en los mismos errores de seguridad, por tanto sugieren que es necesario promover el cambio en los ecosistemas de trabajo en los que se desarrollan los proyectos de software, que se basen en prácticas respaldadas por una investigación formal, para lo cual los autores luego de realizar su investigación y categorización sobre los principales recursos disponibles en la red que facilitasen una guía u orientación general sobre seguridad en el software destacaron 19 recursos, entre ellos múltiples instituciones reconocidas en el área como el SEI, CERN, OWASP e InfoWorld

Hilburn T.B y Humphrey W.S.[9] en su artículo "Teaching teamwork" discuten sobre la importancia de la formación de los ingenieros de software desde el pregrado afirmando que en lugar de enseñarles cómo no se debería programar un software,

---

[8] ACAR, Yasemin; STRANSKY, Christian; WERMKE, Dominik; WEIR, Charles; MAZUREK, Michelle L. y FAHL. Sascha. Developers Need Support, Too: A Survey of Security Advice for Software Developers [En linea]. Leibniz University Hannover, CISPA, Saarland University, Security Lancaster, University of Maryland. IEEE. 2017. pp. 22 – 26.

[9] HILBURN, T.B. y HUMPHREY, W.S. Teaching teamwork in Software [En línea]. IEEE. 2012. pp. 72 – 77.

se enseñe como realmente debería de programarse a través de una metodología formal que les ayude a organizar y planear su trabajo, por tal motivo proponen el uso inicial de cursos introductorios de PSP, para posteriormente hacer uso de TSPi, que es una introducción académica de TSP, que es cíclica compuesta por tres fases, así los ingenieros desde su formación podrán comprender la importancia de planificar su trabajo, realizar un seguimiento de su progreso, administrar la calidad del software y analizar y mejorar su rendimiento, para obtener resultados óptimos en la calidad y seguridad de sus proyectos.

## 2.2 MARCO TEORICO

### 2.2.1 Seguridad Informática

Los constantes cambios a los que está sometido el mundo moderno, la globalización de la economía y el gran flujo de información presente en el diario vivir de la sociedad ha inhibido la necesidad de soportar estas actividades mediante el uso de herramientas que estuvieran a la par con el rápido desarrollo del ecosistema en que se vive, por tanto es asi como las tecnologías de la información y las comunicaciones surgen como la gran alternativa para suplir estos requerimientos de información y desarrollo. Las TIC's han facilitado de una u otra manera la forma en que las personas realizan sus actividades y se adaptan a los constantes cambios, ya que poseen el potencial para cambiar drásticamente a las personas, las organizaciones y las prácticas de negocio.

A medida que el volumen de información sigue incrementándose de forma exponencial, los sistemas de información se han tornado cada vez más complejos y en general la forma en que la tecnología ha revolucionado nuestra manera de interactuar con el mundo, lo que ha puesto de manifiesto una mayor cantidad de vulnerabilidades, derivando en una carrera sin fin por parte de las empresas en

contra de los ciberdelincuentes para proteger su infraestructura tecnológica e información. Es así como la Seguridad Informática busca la protección y conservación de la plataforma tecnológica de una organización, tanto a nivel de hardware, software o de datos, se enfoca en reducir las vulnerabilidades y eventos de seguridad que se pudieran presentar mediante la implementación de técnicas y procedimientos que detecten y permitan controlar a tiempo estos incidentes con miras a facilitar el desarrollo adecuado de los procesos propios de la empresa. Salvaguardando los sistemas de información de accesos, usos, modificaciones o cambios en las configuraciones que no han sido autorizados.

Una de los paradigmas más importantes que persigue la seguridad informática es precisamente la protección de los datos, ya que como lo menciona Miguel Soriano[10] en su artículo "Seguridad en redes y seguridad de la información" su objetivo principal es buscar que la información, que es el activo más importante de una empresa, este siempre disponible, se conserve tal cual como fue concebida y solo sea accesible por el personal autorizado para este fin. De tal forma que ya no solo se habla de este término como una forma o medio de protección de datos sino como una temática fundamental dentro de la cultura organizacional de una empresa dado que todos los miembros del personal hacen parte directa o indirectamente de la custodia de la información.

El portal de seguridad informática SEARCHSECURITY[11] afirma que las amenazas a la información sensible y privada se presentan en diferentes formas, como el malware, ataques de phishing, robo de identidad, ransomware, etc. sugiere que para disuadir a los atacantes y mitigar las vulnerabilidades en varios puntos, es necesario implementar y coordinar múltiples controles de seguridad como parte de una

---

[10] SORIANO, Miguel. Seguridad en redes y seguridad de la información [En línea]. Improvet. Republica Checa. Primera Edición. 2014. [citado 29 junio de 2018]. Disponible en:<http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf>

[11] ROUSE, Margaret. Information security (infosec) [En linea]. Searchsecurity. Techtarget. Septiembre, 2016. [citado 29 junio de 2018]. Disponible en: <https://searchsecurity.techtarget.com/definition/information-security-infosec>

estrategia de defensa, también sugiere que las empresas deberían definir planes de respuesta a incidentes que les permite contener y limitar el daño, para asi eliminar la causa y aplicar controles de defensa actualizados.

Así como lo sugiere Jesús Ramón Jiménez Rojas[12] en su artículo “La Seguridad Informática Y El Usuario Final” cada vez son más las empresas que invierten cuantiosos recursos en la protección de sus activos de información lo que evidencia que la seguridad informática ha denotado ser un punto crítico en cualquier clase de sistema de información sin importar su fin, pero además el autor agrega que no solo se debe pensar en seguridad informática asociada a las grandes plataformas de protección o herramientas de seguridad, sino también en el acceso y uso de los sistemas por parte de los usuarios finales quien según él considera el eslabón más vulnerable en la cadena de custodia de la información, por cuanto son más propensos a acceder a amenazas por sus desconocimiento, por tanto, si bien es cierto pensar por un lado en la infraestructura también es cierto que la seguridad debe dar cobertura a la usuarios finales y a las prácticas que facilitarían una mayor retención y protección de su información.

[12] JIMÉNEZ ROJAS, Jesús Ramon. La seguridad informática y el usuario final [En linea]. Revista Digital Universitaria, UNAM. Abril, 2018. [citado 29 junio de 2018]. Disponible en: <http://www.revista.unam.mx/vol.9/num4/art20/art20.pdf>

### 2.2.2 Principios de la seguridad informática

Los pilares de la seguridad Informática se basan en la forma en que debería ser preservada y administrada la información, como lo menciona Ciro Antonio Dussan Clavijo[13] en su artículo "Políticas de Seguridad Informática" el primero pilar es la integridad, que busca conservar los datos tal cual como fueron concebidos desde su origen hasta su destino, es decir, que estos no hayan sufrido modificaciones o daños en cualquier instante de su custodia; el segundo pilar es la confidencialidad, que define que la información solo puede ser accedida y utilizada por el personal autorizado para tal fin, para esto existen diferentes herramientas como el encriptamiento, protección por contraseñas, etc. Que dificulta el acceso a la información por parte de terceros; el tercer pilar es la disponibilidad, que busca que la información siempre este accesible cuando sea requerida por el usuario. Así mismo Soriano, Miguel [14] define estos pilares como servicios de seguridad afirmando que estos están destinados a garantizar que la información y los sistemas de información pueden tener la protección adecuada ante cualquier eventual transacción o uso de los mismos, agrega además que es necesario contar con otros servicios tales como la autenticación, control de acceso y no repudio para tener un mayor control de la información.

### 2.2.3 Clasificación de la seguridad informática

La seguridad Informática en si es un conjunto de controles, prácticas y herramientas en caminadas a proteger los activos de información, estas por su naturaleza y el área como tal en la que se desempeñan se pueden agrupar en protección física y lógica.

---

[13] DUSSAN CLAVIJO, Ciro Antonio. Políticas de seguridad informática. ENTRAMADO [En línea]. Junio de 2016. [citado 29 junio de 2018]. Disponible en: <http://www.unilibrecali.edu.co/images2/revista-entramado/pdf/pdf_articulos/volumen2/Politicas_de_seguridad_informtica.pdf>

[14]. SORIANO, Miguel. Seguridad en redes y seguridad de la información [En línea]. Improvet. Republica Checa. Primera Edición. 2014. [citado 29 junio de 2018]. Disponible en:<http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf>

La Universidad Internacional de Valencia[15] en su portal web define la seguridad lógica como la protección que se le da a los datos en cualquier software o sistema de información y como se accede a estos, buscando la preservación de los tres pilares de la seguridad informática mencionados con anterioridad. Algunos parámetros con los que debería contar cualquier sistema para garantizar la protección básica de sus datos son: un módulo de autenticación en donde solo el personal registrado y con autorización pueda acceder de tal forma que el usuario autenticado pueda ver únicamente la información para la cual tiene permiso; debe tener niveles de control para la realización de transacciones de datos que garanticen la integridad y recepción de los mismos a través de diferentes métodos como el de encriptamiento, uso de contraseñas, etc; limitación y control de servicios y recursos, es decir que no todos tiene las mismas facultades para manejar la información, habrá quienes solo podrán consultar, otros que podrán modificarla u otros que supervisaran su manipulación.

La seguridad física por su parte como lo menciona David Hutter[16] se refiere a las medidas de protección y control de la infraestructura física donde se encuentran almacenados los sistemas de información, así como también las estaciones de trabajo remoto que tienen acceso a esta. Existen diferentes tipos de incidentes que pueden afectar o poner en riesgos la seguridad física de los sistemas, por un lado, se tienen los incidentes provocados por desastres naturales como terremotos, tormentas, incendios, etc. y por otra parte están los incidentes provocados por acciones del hombre ya sea de forma accidental o intencional que vulneren de manera interna o externa la seguridad.

---

[15] UNIVERSIDAD INTERNACIONAL DE VALENCIA. Conceptos sobre seguridad lógica informática [En linea]. VIU. Marzo, 2018. [citado 29 junio de 2018]. Disponible en: <https://www.universidadviu.com/conceptos-seguridad-logica-informatica/>

[16] HUTTER, David. Physical Security and Why It Is Important [En linea]. SANS Institute InfoSec Reading Room. Junio, 2016. [citado 29 junio de 2018]. Disponible en: <https://www.sans.org/reading-room/whitepapers/physical/physical-security-important-37120>

Para prever este tipo de incidentes es necesario establecer controles y planes de contingencia que salvaguarden y garanticen la continuidad del negocio, tales como establecer controles de acceso físico mediante sistemas de seguridad como algunos mencionados en el documento de "Políticas de seguridad de la información – Seguridad Fisica" del Ministerio del Interior[17] como el uso de sistemas biométricos, utilización de personal de vigilancia, puertas eléctricas, circuitos cerrados de televisión, protección contra amenazas ambientales, mantenimiento de equipos, etc., también se deben realizar capacitaciones con el personal sobre el uso adecuado de sus puestos de trabajo, se deben establecer planes de contingencia como tener réplicas de almacenamiento de la información en diferentes ubicaciones, respaldo del suministro eléctrico para los equipos, planes de atención a desastres, todo esto con el fin de garantizar la prestación del servicio.

### 2.2.4 Ciclo de vida del software

El ciclo de vida de un software es un proceso iterativo y secuencial compuesto de múltiples fases que culminan en la creación de un sistema de información independiente de su fin, estas fases suelen variar de una metodología a otra, pero como lo presenta el autor Motea Alwan[18] las más distintivas son las siguientes:

- Planeación: En esta fase se define un plan de alto nivel preliminar sobre la ejecución prevista del proyecto de software, en donde es necesario conocer la factibilidad del proyecto con el objetivo de definir el alcance del problema y sus posibles soluciones, así mismo estimar recursos, beneficios y demás elementos necesarios para su ejecución.
- Recolección y análisis de requisitos: Esta fase implica analizar de forma detalla y minuciosa los requisitos del negocio provenientes del usuario final, en primera instancia se reúne requisitos funcionales, no funcionales y de

[17] MINISTERIO DEL INTERIOR. Políticas De Seguridad De La Información - Seguridad Física [En linea]. Julio, 2014. [citado 29 junio de 2018]. Disponible en: <http://www.mininterior.gov.co/sites/default/files/oip-2014-psi-especificas-4_seguridad_fisica.doc>
[18] ALWAN, Motea. What is System Development Life Cycle? [En linea]. Airbrake. Enero, 2015. [citado 29 junio de 2018]. Disponible en: <https://airbrake.io/blog/sdlc/what-is-system-development-life-cycle>

seguridad, se crean los diagramas de procesos y luego se realiza un análisis estructurado de la información recopilada, siendo una de las primeras fases y de las más importantes es necesario que se destine el tiempo, energía y los recursos que sean necesarios para una acertada definición de las necesidades del cliente y de la funcionalidad del sistema.

- Diseño: En esta fase de describen en detalle las características y funcionalidades necesarias que satisfarán los requisitos funcionales del sistema que se implementará. Es durante esta fase que se consideraran la estructura de los componentes esenciales, el procesamiento y los procedimientos para alcanzar los objetivos de implementación.
- Desarrollo: Esta es la fase de codificación en donde se toman los diseños y requerimientos definidos en las fases anteriores y se transforman en el sistema, las dos actividades principales que se realizan en esta fase son el desarrollo de la infraestructura de TI que soportaran el sistema a implementar y por otra parte la creación de la base de datos y los programas.
- Integración y pruebas: En esta fase se realiza la integración del sistema con los elementos con los que interactuara cuando este en producción, como también se realizan pruebas a todos los programas y procedimientos desarrollados para determinar si el diseño propuesto cumple con el conjunto inicial de objetivos y alcances establecidos. Otra parte de esta fase es la verificación y validación, que ayudarán a garantizar la finalización exitosa del programa.
- Implementación y mantenimiento: Esta fase implica la instalación y configuración real del sistema y del entorno donde funcionaria; este paso pone en producción el sistema pasando del entorno desarrollo al entorno final. Además, implica el mantenimiento y actualizaciones requeridas a realizarse de forma regular, aquí los usuarios finales pueden ajustar el sistema, si lo desean, para aumentar el rendimiento, agregar nuevas capacidades o cumplir los requisitos adicionales del usuario.

Otra versión es la presentada en la metodología de desarrollo seguro de software propuesta por Gary McGraw[19] en su artículo Software Security, este autor define un conjunto de buenas prácticas de ingeniería de software que propone deben ser aplicadas en cada fase del ciclo de desarrollo independiente del modelo o metodología que se desee aplicar para la implementación del proyecto, estableciendo así un grupo de tareas o actividades genéricas para el desarrollo seguro de software.

McGraw en las diferentes etapas del desarrollo de software hace referencia a ciertas prácticas oportunas a llevar a cabo en cada una de ellas así:

- Etapa de Requerimientos: McGraw sugiere que, en esta fase aparte de los requisitos funcionales del sistema, también se deben especificar requisitos de Seguridad.
- Etapa de diseño: El autor sugiere que se debe documentar todas las suposiciones e identificar cuáles podrían ser los posibles ataques al sistema a desarrollar.
- Etapa de codificación: recomienda el uso de herramientas de análisis estático, para encontrar fallas y vulnerabilidades en el código.
- Etapa de pruebas: Se deben hacer pruebas de seguridad funcional y pruebas de seguridad basadas en riesgos.

[19] MCGRAW, G. Software Security [En línea]. CIGITAL. Marzo, 2004. [citado 29 junio de 2018]. Disponible en: <https://www.cigital.com/papers/download/software-security-gem.pdf>

### 2.2.5 Buenas prácticas en seguridad de la información

COBIT es una modelo que permite evaluar y controlar todos los aspectos de TI de una organización que van desde los sistemas de información hasta el personal que los administra, enfocado siempre a los objetivos del negocio.

Así como se describe en el boletín N° 54 del área de auditoría y control de las Universidad de EAFIT[20] este modelo propone básicamente un entorno donde en un principio se analizan y evalúan los criterios de información de la organización, luego se audita los recursos o infraestructura de TI propios y por último se evalúan los procesos y/o procedimientos de la empresa que interactúan con todos estos factores. Su finalidad es ofrecer a la organización una herramienta que automatice y facilite el cumplimiento de los objetivos del negocio haciendo un uso óptimo de los recursos TI.

ITIL[21] (Information Technology Infrastructure Library) es un conjunto de buenas prácticas para la administración de recursos y servicios de TI, con el objetivo de lograr calidad y eficiencia en los procesos productivos de la organización a través de la mejora continua. ITIL estructura el ciclo de vida de los servicios en cinco etapas:

- La estrategia del servicio se base en la identificación de los activos estratégicos de la empresa, es decir que generan valor o que la diferencia en el mercado.

---

[20] ÁREA DE AUDITORÍA Y CONTROL. COBIT: Modelo para auditoría y control de sistemas de información [En linea]. Universidad EAFIT, Boletín 54. Mayo, 2007. [citado 29 junio de 2018]. Disponible en: <http://www.eafit.edu.co/escuelas/administracion/consultorio-contable/Documents/boletines/auditoria-control/b13.pdf>

[21] RAMÍREZ BRAVO, Pia. y DONOSO JAURES, Felipe. METODOLOGÍA ITIL: Descripción, Funcionamiento y Aplicaciones [En linea]. Facultad de Ciencias Económicas y Administrativas, Universidad De Chile. [citado 29 junio de 2018]. Disponible en: <http://repositorio.uchile.cl/tesis/uchile/2006/donoso_f/sources/donoso_f.pdf>

- El diseño del servicio busca explorar el portafolio de servicios identificando debilidades y fortalezas para así definir estándares y políticas que permitan mejorar la calidad de los mismos.
- La transición del servicio cubre el proceso de cambio entre la mejora de servicios existentes o la creación de nuevos servicios de ser necesario, identificados en el diseño.
- La operación del servicio busca gestionar y controlar los procesos necesarios para la puesta en producción de un servicio.
- La mejora continua del servicio busca optimizar la calidad del servicio ofrecido a través de la retroalimentación constante de la estrategia, diseño, transición y operación que faciliten la adaptación al entorno.

### 2.2.6 Estándares de seguridad y calidad

La ISO 27001 se remonta a la norma BS7799, es un estándar publicado originalmente por Grupo BSI en 1995. Fue escrito por el Gobierno del Reino Unido Departamento de Comercio e Industria (DTI), y consistió en varias partes.

La primera parte, que contiene las mejores prácticas para la gestión de seguridad de la información, se revisó en 1998, después de una larga discusión en los organismos de normalización en todo el mundo, fue finalmente aprobada por la ISO como ISO / IEC 17799, en ese entonces llamado "Tecnología de la información - Código de buenas prácticas para el manejo seguro de la información". Luego en el año 2000 la ISO/IEC 17799 paso por una nueva revisión en junio de 2005 y posteriormente fue incluida en la serie ISO 27000 de normas como es conocida hoy en día la ISO / IEC 27002 en julio del año 2007.

La segunda parte de BS7799 fue publicado por primera vez por BSI en 1999, conocida como BS 7799 Parte 2, titulada "Información sobre los Sistemas de Gestión de Seguridad -. Especificación con orientación para su uso" BS 7799-2 se

centró en cómo implementar un sistema de gestión de seguridad de la información (SGSI), en referencia a la estructura de gestión de seguridad y los controles identificados en BS 7799-2. Más tarde se convirtió en la norma ISO / IEC 27001. La versión 2002 de BS 7799-2 presentó el Plan-Do-Check-Act (PDCA), alineándose con las normas de calidad como la ISO 9000. BS 7799 parte 2 fue adoptado por la ISO como ISO / IEC 27001 en noviembre de 2005.

La tercera parte BS 7799 se publicó en 2005, que abarca el análisis de riesgos y la gestión. Se alinean con la norma ISO / IEC 27001 y es compatible con las normas ISO 9001 e ISO 14001.[22]

La metodología CMMI, como es descrita por el autor Mark C. Paulk[23], data sus orígenes a finales de la década de los años 80 en el departamento de defensa de los Estados Unidos, debido a que tenían muchos problemas con el software que solicitaban desarrollar a otras empresas dado que casi siempre se alargaban los tiempos de entrega y los presupuestos se incrementaban, ante esta problemática convocaron una reunión de expertos que llegaron a la conclusión de que era necesario la creación de un instituto para el desarrollo de software que apoyara al departamento de defensa. Es ahí donde el SEI (Software Engenieering Institute) en 1991 publico el modelo de capacidad de madurez CMM, donde su primera prioridad consistía en identificar las áreas más críticas del desarrollo de software y evaluarlas para determinar procesos de mejora, con el pasar de los años la metodología ha madurado a un conjunto de reglas y buenas prácticas apoyada en una serie de herramientas de software y no software.

---

[22] JBW GROUP INTERNATIONAL INFORMATION ASSURENCE. Evolution of an International Information Security Standard [En linea]. JBWGroup. Abril, 2010. [citado 29 junio de 2018]. Disponible en: <http://www.jbwgroup.com/assets/PDFs/JBW%20Group%20-%20EU%20-%20InfoSec%20History%20V2-N2.1.pdf>

[23] PAULK, Mark. C. A History of the Capability Maturity Model for Software [En Linea]. Carnegie Mellon University. 2009. [citado 29 junio de 2018]. Disponible en: <https://pdfs.semanticscholar.org/6fb0/c324e08698a9e364693151605a74982b487a.pdf>

En la década de los a los años 90 el SEI decidió unificar los modelos de Ingeniería de Software (SW-CMM), de Ingeniería de Sistemas (SE-CMM) y de Desarrollo Integrado de Productos (IPD-CMM) creando en el año 2002 el CMMI (Capability Maturity Model Integration).

Los autores Sakgasit Ramingwong y Lachana Ramingwong[24] datan el origen de la metodologia PSP (Personal Software Process) al año 1993 desarrollada por Watts S. Humphrey, afirmando que PSP es un enfoque disciplinado y estructurado para el desarrollo de software. Mediante el uso de los conceptos y métodos de PSP en su trabajo, los individuos en casi cualquier campo técnico pueden mejorar sus habilidades de estimación y de planificación, hacer compromisos que puedan cumplir, gestionar la calidad de su trabajo, y reducir el número de defectos en sus productos.

Luego de los exitosos resultados de PSP se denoto que era casi imposible mantener la disciplina que se requiere para las prácticas de esta metodología si el entorno que lo rodea no producía ese incentivo y demanda del mismo. Por tanto, Humphrey[25] decidió desarrollar el Team Software Process (TSP) para la unidad operativa más pequeña en la mayoría de las organizaciones, que correspondía al equipo del proyecto. TSP fue realmente diseñado para ser un proceso de nivel 5 de CMMI para equipos de proyectos.

El desarrollo de una guía de buenas prácticas de seguridad para el desarrollo de software basado en estándares reconocidos se fundamenta en el uso de una metodología con un enfoque disciplinado y estructurado como lo es PSP que facilita

---

[24] RAMINGWONG, Sakgasit; RAMINGWONG, Lachana. Implementing a Personal Software Process (PSP SM). Course: A Case Study [En linea]. Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Thailand. Abril, 2012. [citado 29 junio de 2018]. Disponible en: <URL: http://dx.doi.org/10.4236/jsea.2012.58074>

[25] SOFTWARE ENGENIEERING INSTITUTE. The Team Software Process SM (TSP SM) [En linea]. SEI. Noviembre, 2000. [citado 29 junio de 2018]. Disponible en: <https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13754.pdf>

el logro y consecución de las metas, mejorando la productividad, afianzando los buenos hábitos de programación, permitiendo la detección temprana de defectos y riesgos lo que mejora la seguridad y calidad del producto. Además de la integración de todo el equipo del proyecto de desarrollo, como lo describe la metodología TSP.

Por otro parte un apoyo ideal para la estandarización de procesos durante el ciclo de desarrollo de software es CMMI en su nivel de madurez dos que aumenta la eficiencia de las actividades desarrolladas, reduciendo costos mediante la planificación, medición y evaluación de sus procesos. Todas estas prácticas están enmarcadas en los controles de seguridad definidos por la norma ISO 27001 en lo que se refiere a desarrollo y mantenimiento de software.

### 2.2.7 Aplicaciones relacionadas

Algunas aplicaciones exitosas de guías relacionadas directa o indirectamente con el desarrollo seguro de software se relacionan a continuación:

- Dado el crecimiento exponencial sobre el uso de aplicaciones móviles y el amplio espectro del mercado dominado por las aplicaciones de salud, la Agencia de Calidad Sanitaria de Andalucía[26] en España ha propuesto la creación de una Guía de Buenas Prácticas para el desarrollo de aplicaciones móviles relacionadas con la salud, cuyo objetivo es definir un conjunto de criterios y recomendaciones que toda app de este tipo debería cumplir para garantizar la seguridad y calidad de la información suministrada a los usuarios. Este guía cuenta con 31 recomendaciones divididas en cuatro secciones las cuales son: Diseño y pertinencia; Calidad y seguridad de la información; Prestación de servicios y Confidencialidad y Privacidad. Con el objetivo de que las empresas

[26] AGENCIA DE CALIDAD SANITARIA DE ANDALUCÍA, Consejería de Salud. Estrategia de calidad y seguridad en aplicaciones móviles de salud [En Línea]. Andalucía, España, 2012. [citado 29 junio de 2018]. Disponible en: <http://www.calidadappsalud.com>

de desarrollo de apps de salud adoptaran el conjunto de buenas prácticas definidos en la guía la Agencia de Calidad Sanitaria de Andalucía estableció un reconocimiento llamado “Distintivo AppSaludable” que se le otorga a las aplicaciones que cumplan con estos criterios ya sean de origen público o privado o aplicaciones desarrolladas en cualquier parte del mundo no solo en España, dicho reconocimiento es otorgado bajo un debido procedimiento de evaluación de la app y una vez dado le permite a la aplicación ser parte del catálogo de apps que se destacan por sus componentes de seguridad y calidad.

A sí mismo la Agencia de Calidad Sanitaria de Andalucía es una de las organizaciones que conforman el grupo definido por los miembros de la Unión Europea para el desarrollo de las recomendaciones y criterios de calidad que permitan evaluar la salud móvil en los países que conforman esta asociación de países europeos.

- El gobierno de Argentina recientemente realizó una guía de buenas prácticas para el desarrollo de aplicaciones móviles como una iniciativa de la Dirección Nacional de Protección de Datos Personales[27] del Ministerio de Justicia y Derechos Humanos de la Nación que contó con el apoyo especializado del equipo de Seguridad en TIC de la Fundación Sadosky. Esta guía busca tener en cuenta el consentimiento del titular de los datos personales, además incluye una metodología con una serie de pasos para desarrollar y teniendo especial cuidado con las aplicaciones desarrolladas para menores. Sugiere que para que pueda llegar a tener un buen grado de privacidad debe existir primero seguridad, ya que una vez terminado el producto si este no dispone de los lineamientos básicos de seguridad se vuelve una labor muy dispendiosa y además costosa el realizar cualquier arreglo.

[27] DIRECCIÓN NACIONAL DE PROTECCIÓN DE DATOS PERSONALES. Guía de buenas prácticas para el desarrollo de Apps [En línea]. Ministerio de Justicia y Derechos Humanos, Argentina, abr. 2015 [citado 29 junio de 2018]. Disponible en: <http://www.jus.gob.ar/media/3075908/guiabpsoftware.pdf>

- Un estándar reconocido internacionalmente para la definición de las propiedades, requisitos y objetivos de seguridad para recursos de TI es la ISO/IEC 15408 definido por La Organización Internacional para a Estandarización[28], este estándar establece dos formas de identificar los requisitos funcionales y de seguridad de las tecnologías de la información, la primera es la construcción del Perfil de Protección (PP) en la cual se desarrolla un documento con las características de seguridad que se desea tenga el software o producto, en el cual se realiza un listado de todos aquellos requisitos de seguridad; esta fase suele estar asociada con la toma de requerimientos o el diseño del software e inicia con la identificación del entorno de trabajo donde funcionara el desarrollo, allí se busca identificar cual va a ser la funcionalidad del producto que se busca desarrollar, bajo qué condiciones deberá trabajar, los activos de información relacionados que requieren protección y a que amenazas están expuestos, para finalmente definir los requerimientos de seguridad necesarios que permitan mitigar o evitar los riesgos generados por las amenazas identificadas. La segunda forma definida por este estándar corresponde a los Objetivos de Seguridad que define los requisitos y funciones de seguridad que deberían ser evaluados y tenidos en cuenta durante todo el ciclo de producción para asegurar que el producto cumpla con los parámetros de seguridad identificados.

[28] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology, Security techniques, Evaluation criteria for IT security, Part 1: Introduction and general model ISO. ISO/IEC 15408-1:2005. Suiza: ISO. 1999

# 3 METODOLOGIA DE INVESTIGACION

## 3.1 TIPO DE INVESTIGACION

El estudio desarrollado es de tipo descriptivo bajo el enfoque cuantitativo, ya que se trata de observar y medir las variables relacionadas con la seguridad en el desarrollo de software.

## 3.2 TECNICAS DE RECOLECCION DE DATOS

El proceso de obtención de información se fundamenta en el análisis y síntesis de las buenas prácticas de desarrollo seguro de software identificadas en los estándares y metodologías seleccionados para el estudio (PSP/TSP, CMMI Nivel de madurez 2 y la norma ISO 27001), cada una de las cuales cuenta con un documento oficial y guía que describe y desarrollo cada una de las metodologías.

## 3.3 TECNICAS DE PROCESAMIENTO DE DATOS

La metodología para el procesamiento de datos utilizada se base en la construcción de indicadores que faciliten el proceso de categorización de las mejores prácticas de desarrollo seguro de software identificadas en los estándares y metodologías seleccionados.

## 3.4 METODOLOGÍA DE DESARROLLO

El proyecto se desarrolló en tres fases:

1. Fase de fundamentación y análisis: En esta fase se pretende realizar la identificación y construcción de los indicadores y criterios que permitan evaluar de forma asertiva las diferentes prácticas de seguridad en el desarrollo de software descritas por las metodologías seleccionadas.
2. Fase de evaluación y selección: con los criterios identificados para el análisis del conjunto de prácticas seguras para el desarrollo de software se evaluaron y caracterizaron cuales son más importantes y de fácil integración con las otras metodologías, realizando la selección de las mejores prácticas.
3. Fase de implementación: Luego de identificar el conjunto de prácticas se procedió a identificar posibles factores de riesgo que pudieran derivarse de su implementación, finalmente, se procedió a realizar la creación de la guía de buenas practicas seleccionadas.

# 4 RESULTADOS

## 4.1 BUENAS PRACTICAS PARA EL DESARROLLO DE SOFTWARE

A continuación, se describen en detalle las practicas relaciones con el desarrollo seguro de software identificadas durante el análisis realizado a los diferentes estándares y metodologías materia de estudio en el presente documento, además se presenta la relación de cada práctica con el ciclo de vida del software y su correspondiente justificación como practica segura de desarrollo.

### 4.1.1 NTC-ISO/IEC 27001

A continuación, se presentan las practicas seleccionadas en el estándar ISO 27001 que tiene alguna relación o pueden aplicarse al desarrollo de software, describiendo la sección correspondiente a la que pertenecen e indicando la práctica identificada, además se conserva el orden en que son tratadas en la guía oficial de la norma.

Cuadro 1. Practica 1 ISO 27001

| **Ítem** | PRA-ISO-1 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.10.1 Procedimientos operacionales y responsabilidades<br>Objetivo: Busca garantizar el correcto funcionamiento de los sistemas de procesamiento de la información<br>Control: **A.10.1.2 Gestión del Cambio**<br>Observaciones: Evidenciar si se tiene control de los cambios efectuados en los sistemas de procesamiento de información conservando y administrando los de cambios. | | |
| **Fase Ciclo SW** | Desarrollo | | |
| **Justificación** | Durante el desarrollo de software se deben identificar los elementos que deben estar bajo control para restringir y documentar los cambios realizados sobre estos, a su vez dichos cambios o versiones del desarrollo deben ser auditados o revisados para garantizar su completitud e integridad. | | |

Cuadro 2. Practica 2 ISO 27001

| **Ítem** | PRA-ISO-2 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.10.1 Procedimientos operacionales y responsabilidades<br>Objetivo: Busca garantizar el correcto funcionamiento de los sistemas de procesamiento de la información<br>Control: **A.10.1.4 Separación de las instalaciones de desarrollo, ensayo y operación.**<br>Observaciones: Evidenciar si la organización cuenta con entornos separados de desarrollo, pruebas y producción, conservando la similitud de configuración y recursos para que se trabaje de forma uniforme en cada entorno. Con el objetivo de disminuir cambios no deseados en el entorno de producción. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | En el ciclo de vida del software es necesario disponer de entornos de desarrollo separados que permite realizar las pruebas de funcionamiento correspondientes a nuevas implementaciones sin afectar el funcionamiento de las existentes y reduciendo el riesgo de modificaciones o cambios inesperados durante todas las fases. | | |

Cuadro 3. Practica 3 ISO 27001

| **Ítem** | PRA-ISO-3 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A. 10.3 Planificación y aceptación del sistema<br>Objetivo de Control: Reducir significativamente el riesgo de aparición de una falla.<br>Control: **A.10.3.1 Gestión de la capacidad.**<br>Observaciones: Se deben monitorear y administrar los recursos del sistema, estimando los requisitos de capacidad futura para garantizar el correcto funcionamiento de los sistemas. | | |
| **Fase Ciclo SW** | Requerimientos y diseño | | |
| **Justificación** | Durante la fase de diseño se deben estimar los recursos necesarios para que el sistema funcione de forma óptima sin superar los recursos disponibles. | | |

Cuadro 4. Practica 4 ISO 27001

| **Ítem** | PRA-ISO-4 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A. 10.3 Planificación y aceptación del sistema<br>Objetivo de Control: Reducir significativamente el riesgo de aparición de una falla.<br>Control: **A.10.3.2 Aceptación del sistema.**<br>Observaciones: Se deben establecer criterios de aceptación de nuevas implementaciones o actualización de las existentes, además de realizar las respectivas pruebas de funcionamiento durante su desarrollo, previo a su aprobación. | | |
| **Fase Ciclo SW** | Desarrollo, pruebas, despliegue y mantenimiento | | |
| **Justificación** | Durante el desarrollo es necesario aplicar las pruebas suficientes que garanticen el correcto funcionamiento del sistema en cada fase, con el objetivo de lograr cumplir con los criterios de aceptación y poder pasar a la etapa de producción. | | |

Cuadro 5. Practica 5 ISO 27001

| **Ítem** | PRA-ISO-5 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.10.4 Protección contra códigos maliciosos y móviles<br>Objetivo de Control: Garantizar la integridad de la información y proteger el software.<br>Control: **A.10.4.1 Controles contra códigos maliciosos.**<br>Observaciones: Establecimiento de controles para la identificación y prevención de ataques por código malicioso y procedimientos para su recuperación. | | |
| **Fase Ciclo SW** | Desarrollo | | |
| **Justificación** | Durante el desarrollo es necesario definir y codificar procedimientos para la prevención de ataques por códigos maliciosos, de tal forma que la información que será ingresada al sistema sea filtrada previamente antes de ser almacenada, en especial cuando refiere a consultas en la base de datos. | | |

Cuadro 6. Practica 6 ISO 27001

| **Ítem** | PRA-ISO-6 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.11.2 Gestión del acceso de usuarios<br>Objetivo de Control: Garantizar el acceso al sistema solamente de los usuarios autorizados, denegar cualquier otro acceso no autorizado.<br>Control: **A.11.2.1 Registro de Usuarios.**<br>Observaciones: Se deben establecer procedimientos formales para el registro y retiro de usuarios en los sistemas de información | | |
| **Fase Ciclo SW** | Diseño, desarrollo | | |
| **Justificación** | Durante el desarrollo es necesario considerar una adecuada administración de usuarios que permita el acceso solo a las personas registradas en el sistema. | | |

Cuadro 7. Practica 7 ISO 27001

| **Ítem** | PRA-ISO-7 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.11.2 Gestión del acceso de usuarios<br>Objetivo de Control: Garantizar el acceso al sistema solamente de los usuarios autorizados, denegar cualquier otro acceso no autorizado.<br>**Control: A.11.2.2 Gestión de privilegios.**<br>Observaciones: Se deben controlar y gestionar la asignación y uso de los privilegios de usuarios en el sistema. | | |
| **Fase Ciclo SW** | Diseño | | |
| **Justificación** | Durante el diseño se deben identificar y especificar de forma clara todos los actores que intervendrán en el software y que solo puedan desempeñar el rol para el cual fue asignado, restringiendo cualquier otra acción no permitida. | | |

Cuadro 8. Practica 8 ISO 27001

| **Ítem** | PRA-ISO-8 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.1 Requisitos de seguridad de los sistemas de información<br>Objetivo de Control: Incluir el componente de seguridad como parte integral de cualquier sistema de información.<br>Control: A.12.1.1 **Análisis y especificación de los requisitos de seguridad**<br>Observaciones: Para toda nueva implementación de nuevos sistemas o actualización de alguno existente se deben especificar los requerimientos de seguridad necesarios. | | |
| **Fase Ciclo SW** | Toma de requerimientos | | |

Cuadro 8. (Continuación)

| | |
|---|---|
| **Justificación** | Durante el proceso de toma y especificación de requerimientos es indispensable para un desarrollo seguro establecer los requerimientos de seguridad necesarios para proteger la integridad del software y la información, garantizando un adecuado funcionamiento ante la presencia del algún incidente de seguridad, además deben estar estrechamente relacionados con las políticas de seguridad del cliente. |

Cuadro 9. Practica 9 ISO 270001

| **Ítem** | PRA-ISO-9 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.2 Procesamiento correcto en las aplicaciones<br>Objetivo de Control: Prevenir un uso inapropiado de la información, así como evitar la presencia de fallos o modificaciones no autorizados en los sistemas de información o aplicaciones.<br>Control: **A.12.2.1 Validación de los datos de entrada.**<br>Observaciones: Validación de la correctitud e integridad de la información ingresada en la aplicación. | | |
| **Fase Ciclo SW** | Desarrollo, pruebas | | |
| **Justificación** | Durante el desarrollo es necesario establecer controles para filtrar y controlar la información que ingresa al aplicativo de forma que solo ingrese información valida, descartando toda información errada o que contenga códigos maliciosos. | | |

Cuadro 10. Practica 10 ISO 27001

| **Ítem** | PRA-ISO-10 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.2 Procesamiento correcto en las aplicaciones<br>Objetivo de Control: Prevenir un uso inapropiado de la información, así como evitar la presencia de fallos o modificaciones no autorizados en los sistemas de información o aplicaciones.<br>Control: **A.12.2.2 Control de procesamiento interno.**<br>Observaciones: Definir controles de validación para identificar cualquier corrupción de la información derivada de fallos en su procesamiento o acciones malintencionadas. | | |
| **Fase Ciclo SW** | Desarrollo | | |
| **Justificación** | Es necesario validar que la información que se encontrara almacenada en el aplicativo se conserve integra y corresponda a la ingresada en forma y contenido. | | |

Cuadro 11. Practica 11 ISO 27001

| **Ítem** | PRA-ISO-11 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.2 Procesamiento correcto en las aplicaciones<br>Objetivo de Control: Prevenir un uso inapropiado de la información, así como evitar la presencia de fallos o modificaciones no autorizados en los sistemas de información o aplicaciones.<br>Control: **A.12.2.3 Integridad del mensaje.**<br>Observaciones: Identificar y establecer los requisitos necesarios para garantizar la veracidad e integridad de los mensajes en la aplicación, implementando los controles necesarios. | | |
| **Fase Ciclo SW** | Toma de requerimientos, diseño | | |
| **Justificación** | Durante la fase de toma de requerimientos y de diseño se deben identificar qué clase de controles o métodos se implementarán para las transferencias de información, por ejemplo, encriptando los mensajes. Estos controles a implementar durante el desarrollo deben poder garantizar que la información enviada y recibida se conserva tal cual como fue concebida y sea recibida por su destinatario inicial. | | |

Cuadro 12. Practica 12 ISO 27001

| **Ítem** | PRA-ISO-12 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.2 Procesamiento correcto en las aplicaciones<br>Objetivo de Control: Prevenir un uso inapropiado de la información, así como evitar la presencia de fallos o modificaciones no autorizados en los sistemas de información o aplicaciones.<br>Control: **A.12.2.4 Validación de los datos de salida**.<br>Observaciones: Se deben validar los datos de salida de la aplicación, para garantizar que el procesamiento de la información almacenada en la aplicación sea correcto y que sea la espera. | | |
| **Fase Ciclo SW** | Desarrollo, pruebas | | |
| **Justificación** | Los datos que ingresan y son procesados por la aplicación deben ser controlados y revisados antes de salir de la aplicación, para ello es necesario que el sistema tenga controles de la validación que permitan constatar la consistencia y veracidad de los datos generados. | | |

Cuadro 13. Practica 13 ISO 27001

| **Ítem** | PRA-ISO-13 | **Metodología** | ISO 27001 |
|---|---|---|---|
| **Descripción** | A.12.4 Seguridad de los archivos del sistema<br>Objetivo de Control: Se debe garantizar la seguridad de los archivos del sistema.<br>Control: **A.12.4.2 Protección de los datos de prueba del sistema.**<br>Observaciones: Los datos de prueba se deben pasar por un proceso idóneo de selección según las especificaciones del sistema, estos datos deben ser controlados y resguardados | | |
| **Fase Ciclo SW** | Pruebas | | |
| **Justificación** | En la fase de pruebas un componente decisivo para garantizar el funcionamiento correcto y esperado del sistema depende directamente de los datos de prueba seleccionados, estos deben guardar concordancia con la información real que será ingresada y debe considerar múltiples escenarios de uso que garanticen la integridad de la información procesada | | |

### 4.1.2 CMMI Nivel 2

A continuación, se describen las practicas seleccionadas en el estándar CMMI nivel de madurez dos que tienen alguna relación o pueden aplicarse al desarrollo de software, describiendo la sección correspondiente a la que pertenecen e indicando la práctica identificada, además se conserva el orden en que son tratadas en la guía oficial de la metodología. Es necesario recordar que en el nivel de madurez dos o gestionado solo abarca siete áreas de procesos de CMMI[29] correspondientes a esta etapa que son: gestión de requisitos, planificación del proyecto, seguimiento y control del proyecto, gestión de acuerdos con proveedores, medida y análisis, medidas de calidad en el proceso y el producto y gestión de configuración.

[29] Software Engenieering Institute. (noviembre de 2010). CMMI® para Desarrollo, Versión 1.3. SEI. Pag. 49. Recuperado de https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_28782.pdf

Cuadro 14. Practica 1 CMMI Nivel 2

| **Ítem** | PRA-CMM-1 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Configuración<br>Propósito: Busca definir y conservar la integridad de los productos de trabajo, identificando los elementos de configuración, gestionando la configuración, controlando los cambios de la configuración y supervisando la configuración.<br>Meta Especifica: SG 1 Establecer líneas base<br>Objetivo: Establecer las linceas bases de los elementos de configuración identificados<br>Practica Especifica: **SP 1.1 Identificar los elementos de configuración**<br>Subprácticas:<br>1. Identificar los elementos de configuración y sus productos de trabajo<br>2. Asignar identificadores a los elementos de configuración.<br>3. Identificar las características más relevantes de cada elemento<br>4. Especificar cuando entra a ser gestionado cada elemento de la configuración identificado.<br>5. Definir relaciones entre elementos. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Durante todo el ciclo de desarrollo de software un componente fundamental presente en cada fase son los elementos de configuración necesarios para ejecutar los procesos, ya sea por ejemplo en la fase de toma de requerimientos en donde muchas veces requerimos de preparar lista de chequeo, fichas técnicas, etc; en la fase de diseño donde las descripciones de las interfaces o los diferentes diagramas representan también elementos de configuración; en el desarrollo determinar el lenguajes de programación en que se programara, las herramientas que se utilizaran e inclusive el mismo código fuente representan un elemento de configuración necesario para garantizar la realización y obtención de las metas en cada proceso. | | |

Cuadro 15. Practica 2 CMMI Nivel 2

| **Ítem** | PRA-CMM-2 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Configuración<br>Propósito: Busca definir y conservar la integridad de los productos de trabajo, identificando los elementos de configuración, gestionando la configuración, controlando los cambios de la configuración y supervisando la configuración.<br>Meta Especifica: SG 1 Establecer líneas base<br>Objetivo: Establecer las linceas bases de los elementos de configuración identificados<br>Practica Especifica: **SP 1.2 Establecer un sistema de gestión de configuración**<br>Subpracticas:<br>1. Definir metodología para administrar diferentes niveles de gestión<br>2. Proporcionar controles de acceso autorizado al sistema<br>3. Almacenar y recuperar los elementos de configuración en el sistema<br>4. Compartir los elementos de configuración identificados entre los niveles de gestión<br>5. Almacenar y recuperar los registros de gestión<br>6. Generar informes de gestión de configuración desde el sistema de gestión<br>7. Preservar la información del sistema de gestión<br>8. Realizar las modificaciones necesarias a la estructura de gestión | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Durante todo el ciclo de desarrollo de software para los elementos que requieren una configuración es necesario generar un sistema de gestión que administre de forma efectiva y eficiente la configuración de los mismos durante el desarrollo. | | |

Cuadro 16. Practica 3 CMMI Nivel 2

| **Ítem** | PRA-CMM-3 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Configuración<br>Propósito: Busca definir y conservar la integridad de los productos de trabajo, identificando los elementos de configuración, gestionando la configuración, controlando los cambios de la configuración y supervisando la configuración.<br>Meta Especifica: SG 2 Seguir y controlar los cambios<br>Objetivo: Control y supervisión de los elementos de configuración que se encuentran gestionados.<br>Practica Especifica: **SP 2.1 Seguir las peticiones de cambio**<br>Subpracticas:<br>1. Registrar las peticiones de cambio en una base de datos de peticiones<br>2. Analizar el impacto de los cambios propuesto y las posibles correcciones a los mismos.<br>3. Clasificar las peticiones de cambio y asignarles prioridades.<br>4. Hacer seguimiento de las peticiones hasta su cierre. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Toda petición o intención de modificación de un elemento de configuración en cualquier fase del desarrollo debe ser previamente registrada y analizada con el fin de reducir el impacto y mejorar su utilidad. | | |

Cuadro 17. Practica 4 CMMI Nivel 2

| **Ítem** | PRA-CMM-4 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Configuración<br>Propósito: Busca definir y conservar la integridad de los productos de trabajo, identificando los elementos de configuración, gestionando la configuración, controlando los cambios de la configuración y supervisando la configuración.<br>Meta Especifica: SG 2 Seguir y controlar los cambios<br>Objetivo: Control y supervisión de los elementos de configuración que se encuentran gestionados.<br>Practica Especifica: **SP 2.2 Controlar los elementos de configuración** | | |

Cuadro 17. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas:<br>1. Administrar los cambios en los elementos de configuración durante toda la vida dl producto.<br>2. Gestionar las autorizaciones necesarias antes de ingresar una modificación de un elemento de configuración al sistema de gestión de la configuración.<br>3. Revisar los elementos de entrada y salida en el sistema de gestión para ingresar nuevos cambios con el fin de garantizar que se conserve la integridad de los elementos de configuración.<br>4. Realizar seguimiento para garantizar que los cambios no hayan comprometido la seguridad del sistema<br>5.Regsitrar los cambios y su justificación |
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Cuando se va generar una modificación a un elemento de configuración en el sistema se deben administrar los cambios y realizar seguimiento a los mismo con el fin de que estos no comprometan la seguridad de los mismos. |

Cuadro 18. Practica 5 CMMI Nivel 2

| **Ítem** | PRA-CMM-5 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 1 Alinear las actividades de medición y análisis<br>Objetivo: Los objetivos y análisis de medición deben estar correlacionados con las necesidades de información identificadas<br>Practica Especifica: **SP 1.1 Establecer los Objetivos de Medición**<br>1. Documentar las necesidades existentes de información y los objetivos<br>2. Priorizar las necesidades, no siempre es posible o deseable medir inicialmente todas las necesidades de información.<br>3.Documentar, monitorear y actualizar los objetivos de medición, es importante que los usuarios involucrados en los resultados de la medición estén presentes en la definición de los objetivos de medición<br>4. Suministrar retroalimentación cuando sea necesario para afianzar y confirmar las necesidades de información y los objetivos.<br>5. Conservar la trazabilidad entre los objetivos de medición y las necesidades de información identificadas | | |

Cuadro 18. (Continuación)

| | |
|---|---|
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | La medición y análisis es un proceso transversal a todo el ciclo de vida del producto ya que en todas las etapas es necesario cuantificar y analizar las diferentes necesidades de información y objetivos propuestos con miras a asegurar si se están cumpliendo, si son de calidad y si se pueden mejorar los procesos. |

Cuadro 19. Practica 6 CMMI Nivel 2

| **Ítem** | PRA-CMM-6 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 1 Alinear las actividades de medición y análisis<br>Objetivo: Los objetivos y análisis de medición deben estar correlacionados con las necesidades de información identificadas<br>Practica Especifica: **SP 1.2 Especificar las medidas**<br>1. Identificar las medidas ideales en base a los objetivos de medición, estas medidas se clasifican y se especifican por nombre y unidad de medida.<br>2. Conservar la trazabilidad entre las medidas y los objetivos de medición.<br>3. Identificar las medidas existentes que ya se aplican a los objetivos<br>4. Establecer las definiciones operativas para las medidas, es decir, que se ha medido, como se ha medido, se podría obtener el mismo resultado repitiendo la medición<br>5. Clasificar, monitorear y actualizar las medidas, las especificaciones propuestas para las medidas se deben revisar con las partes interesadas para poder determinar si son adecuadas. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Con la definición de medidas acordes a los objetivos de medición se pueden obtener estimaciones proporcionales a los elementos a medir, pudiendo determinar por ejemplo el tiempo estimado de un proyecto o fase, el coste y esfuerzo horas hombre en el desarrollo de un sistema, detección de defectos en todo el ciclo de vida del producto, etc. | | |

Cuadro 20. Practica 7 CMMI Nivel 2

| **Ítem** | PRA-CMM-7 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 1 Alinear las actividades de medición y análisis<br>Objetivo: Los objetivos y análisis de medición deben estar correlacionados con las necesidades de información identificadas<br>Practica Especifica: **SP 1.3 Especificar los procedimientos de recogida y de almacenamiento de datos**<br>1. identificar las fuentes que puedan proporcionar información relevante.<br>2. Identificar para que medidas no cuentan con datos disponibles<br>3. Establecer cómo se van recoger y almacenar los datos para cada medida<br>4. Definir técnicas de recolección de datos<br>5. Soportar los procedimientos de recolección de datos<br>6. Categorizar, monitorear y actualizar los procedimientos de recolección y almacenamiento.<br>7. Actualización de las medidas y objetivos de medición según sea requerido. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | De un procedimiento adecuado de toma y recolección de datos depende si se pueden medir correctamente los objetivos de medición a alcanzar. Por ende para poder satisfacer las necesidades de medición de cada fase del ciclo de vida del producto es requerido definir los procedimientos que más fácil se adapten a las condiciones en que se encuentra la información y como debe ser almacenada. | | |

Cuadro 21. Practica 8 CMMI Nivel 2

| **Ítem** | PRA-CMM-8 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 1 Alinear las actividades de medición y análisis<br>Objetivo: Los objetivos y análisis de medición deben estar correlacionados con las necesidades de información identificadas<br>Practica Especifica: **SP 1.4 Especificar los procedimientos de análisis** | | |

Cuadro 21. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas:<br>1. Definir y priorizar que análisis se van a realizar y que informes se deben presentar.<br>2. Seleccionar las herramientas adecuadas para analizar los datos.<br>3. Establecer los procedimientos de índole administrativa para el estudio y comunicación de los datos.<br>4. Administrar el formato propuesto para el análisis y los informes.<br>5. Evaluar la utilidad de los resultados del análisis. |
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Los resultados de las mediciones deben ser analizados para poder determinar si cumplen con lo espero en los objetivos propuestos y si sirven como punto de partida para definir y mejorar la calidad de los procesos en cada etapa del ciclo de vida del producto |

Cuadro 22. Practica 9 CMMI Nivel 2

| **Ítem** | PRA-CMM-9 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 2 Proporcionar los resultados de medición<br>Objetivo: Se generan los resultados o productos de la medición que están relacionados con las necesidades de información y objetivos identificados<br>Practica Especifica: **SP 2.1 Obtener los datos de la medición**<br>Subpracticas<br>1. Se obtienen los datos para la generación de las líneas base, estos pueden estar almacenados en los proyectos o cualquier otro elemento de la empresa<br>2. Se procede a generar la información necesaria para cualquier otra medida derivada o necesaria.<br>3. Efectuar los procedimientos necesarios para verificar la integridad y autenticidad de los datos. En estas comprobaciones de las mediciones deben ser identificados los errores en la especificación y registro de datos | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | La obtención de todos los datos necesarios para la medición y su posterior validación durante el desarrollo de un proyecto, permiten ayudar a monitorear de formas efectiva el progreso y rendimiento del mismo. | | |

Cuadro 23. Practica 10 CMMI Nivel 2

| **Ítem** | PRA-CMM-10 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 2 Proporcionar los resultados de medición<br>Objetivo: Se generan los resultados o productos de la medición que están relacionados con las necesidades de información y objetivos identificados<br>Practica Especifica: **SP 2.2 Analizar los datos de medición**<br>Subpracticas<br>1. Realizar análisis inicial, interpretar y concluir los resultados preliminares, para esto es necesario definir criterios para entender las conclusiones de la medición.<br>2. Realizar mediciones o análisis complementarios que sean necesarios y prepararlos para ser entregados.<br>3. Realizar la revisión de los resultados preliminares con todas las partes interesadas<br>4. Concretar criterios para análisis posteriores. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Posterior a la obtención y análisis de los datos medidos, es necesario interpretar los resultados para implementar acciones que mejoren el proceso. | | |

Cuadro 24. Practica 11 CMMI Nivel 2

| **Ítem** | PRA-CMM-11 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 2 Proporcionar los resultados de medición<br>Objetivo: Se generan los resultados o productos de la medición que están relacionados con las necesidades de información y objetivos identificados<br>Practica Especifica: **SP 2.3 Almacenar los datos y los resultados**<br>Subpracticas<br>1. Revisar la integridad, precisión y actualización de la información<br>2. Almacenar la información de las mediciones y análisis conforme el procedimiento general de almacenamiento de datos.<br>3. Restringir el acceso a los datos almacenados solo a personal autorizado.<br>4. Gestionar y dar buen uso de la información almacenada | | |

Cuadro 24. (Continuación)

| | |
|---|---|
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | El almacenamiento de todos los datos históricos de la medición servirá de base para fundamentar la implementación de criterios y mediciones futuras, además también permite mejorar el proceso de ejecución del proyecto basado en experiencias pasadas. |

Cuadro 25. Practica 12 CMMI Nivel 2

| **Ítem** | PRA-CMM-12 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Medición y análisis<br>Propósito: Generar la capacidad de medición utilizada para soportar las necesidades de información de la gerencia<br>Meta Especifica: SG 2 Proporcionar los resultados de medición<br>Objetivo: Se generan los resultados o productos de la medición que están relacionados con las necesidades de información y objetivos identificados<br>Practica Especifica: **SP 2.4 Comunicar los resultados**<br>Subpracticas<br>1. Conservar informadas a todas las partes de forma oportuna<br>2. Ayudar a la compresión de los resultados. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Es necesario que una vez realizada la medición se realice la presentación y sustentación de los resultados obtenidos con todos los miembros implicados en todas las fases del desarrollo, para que sean partícipes del proceso de mejora. | | |

Cuadro 26. Practica 13 CMMI Nivel 2

| **Ítem** | PRA-CMM-13 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 1 Monitorización del proyecto frente al plan<br>Objetivo: Se gestiona el rendimiento real del proyecto versus el plan del proyecto<br>Practica Especifica: **SP 1.1 Monitorizar los parámetros de planificación del proyecto** | | |

Cuadro 26. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas<br>1. Gestionar el avance del proyecto frente a lo establecido en el calendario.<br>2. Gestionar el costo y esfuerzo real empleado frente al planeado y determinar las desviaciones significativas.<br>3. Gestionar los parámetros o atributos de cada producto para determinar su desarrollo e identificar las desviaciones significativas frente a lo planeado.<br>4. Gestionar los recursos<br>5. Gestionar la capacidad del personal del proyecto<br>6. Documentar las desviaciones significativas. |
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Con miras a la optimización de recursos y al cumplimiento de los objetivos propuesto en el proyecto de desarrollo, es necesario monitorear cada aspecto y parámetro del mismo, y compáralo con lo presupuestado inicialmente, de tal forma que este se ajuste a lo que se había planeado y de haber algún desvió significativo se puedan aplicar las respectivas acciones correctivas. |

Cuadro 27. Practica 14 CMMI Nivel 2

| **Ítem** | PRA-CMM-14 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 1 Monitorización del proyecto frente al plan<br>Objetivo: Se gestiona el rendimiento real del proyecto versus el plan del proyecto<br>Practica Especifica: **SP 1.2 Monitorizar los compromisos**<br>Subpracticas<br>1. Revisar los compromisos de forma periódica<br>2. Identificar cuales compromisos no se están cumpliendo<br>3. Documentar los resultados. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Revisar constantemente los compromisos adquiridos para el desarrollo del proyecto y nivel de cumplimiento, facilita la obtención de las metas propuestas y así mejorar la calidad del proyecto | | |

Cuadro 28. Practica 15 CMMI Nivel 2

| **Ítem** | PRA-CMM-15 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 1 Monitorización del proyecto frente al plan<br>Objetivo: Se gestiona el rendimiento real del proyecto versus el plan del proyecto<br>Practica Especifica: **SP 1.3 Monitorizar los riesgos del proyecto**<br>Subpracticas<br>1. Revisar de forma periódica el estado y la documentación de los riesgos identificados en el proyecto<br>2. Actualizar la documentación de los riesgos conforme ocurra algún cambio<br>3. Comunicar del estado de los riesgos a todo el personal relacionado | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Monitorear los riesgos identificados y revisar de forma constante su evolución, para gestionar y controlar de forma segura los incidentes de seguridad que se pudiesen presentar. | | |

Cuadro 29. Practica 16 CMMI Nivel 2

| **Ítem** | PRA-CMM-16 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 1 Monitorización del proyecto frente al plan<br>Objetivo: Se gestiona el rendimiento real del proyecto versus el plan del proyecto<br>Practica Especifica: **SP 1.6 Llevar a cabo las revisiones del progreso**<br>Subpracticas<br>1. Comunicar con frecuencia el estado de las actividades y productos del proyecto a las partes interesadas.<br>2. Revisar y analizar las mediciones realizadas<br>3. Identificar y documentar las desviaciones significativas frente a lo planeado<br>4. Documentar las solicitudes de cambio para los productos y procesos<br>5. Documentar los resultados<br>6. Gestionar las peticiones de cambio hasta su cierre | | |

Cuadro 30. (Continuación)

| | |
|---|---|
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Aplicar mediciones a los productos y procesos del proyecto en diferentes momentos de su desarrollo, permite controlar el estado, calidad y rendimiento del mismo. |

Cuadro 30. Practica 17 CMMI Nivel 2

| **Ítem** | PRA-CMM-17 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 1 Monitorización del proyecto frente al plan<br>Objetivo: Se gestiona el rendimiento real del proyecto versus el plan del proyecto<br>Practica Especifica: **SP 1.7 Llevar a cabo las revisiones de hitos**<br>Subpracticas<br>1. Revisar los hitos más significativos con las partes interesadas en determinados momentos del calendario.<br>2. Realizar una revisión de compromisos, estados, planes y riesgos del desarrollo del proyecto.<br>3. Identificar los temas más relevantes y sus respectivos impactos<br>4. Documentar los resultados<br>5. Gestionar las acciones hasta su cierre | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | El hecho de no tratar una cuestión o problema no identificado por falencias en la revisión podría representar errores de procesamiento o vulnerabilidades que podrían ser explotadas. | | |

Cuadro 31. Practica 18 CMMI Nivel 2

| **Ítem** | PRA-CMM-18 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 2 Gestionar las acciones correctivas hasta su cierre<br>Objetivo: Gestionar las acciones correctivas hasta su cierre, derivadas de un rendimiento o resultado con una desviación relevante a la planeada. | | |

Cuadro 32. (Continuación)

| | |
|---|---|
| **Descripción** | Practica Especifica: **SP 2.1 Analizar las cuestiones**<br>Subpracticas<br>1. Identificar todos los temas que requieran ser analizados.<br>2. Analizar los temas y determinar si requieren acciones correctivas. |
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Analizar las cuestiones más relevantes que requieran tratamiento y determinar si requieren tratamiento facilitara la prevención y aparición de incidentes. |

Cuadro 32. Practica 19 CMMI Nivel 2

| **Ítem** | PRA-CMM-19 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 2 Gestionar las acciones correctivas hasta su cierre<br>Objetivo: Gestionar las acciones correctivas hasta su cierre, derivadas de un rendimiento o resultado con una desviación relevante a la planeada.<br>Practica Especifica: **SP 2.2 Llevar a cabo las acciones correctivas**<br>Subpracticas<br>1. Definir y documentar las acciones necesarias para trata las cuestiones identificadas.<br>2. Definir acuerdos sobre las acciones a tomar entre todas las partes interesadas.<br>3. Gestionar los cambios necesarios a los compromisos tanto internos como externos | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Definir acciones correctivas para mitigar las cuestiones que requieran atención, mejorara la calidad y seguridad del proyecto evitando en gran medida la aparición futura de incidentes. | | |

Cuadro 33. Practica 20 CMMI Nivel 2

| **Ítem** | PRA-CMM-20 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Monitorización y control del proyecto<br>Propósito: Gestionar el desarrollo del proyecto para aplicar acciones correctivas cuando el rendimiento no cumple con un nivel aceptable según lo planeado.<br>Meta Especifica: SG 2 Gestionar las acciones correctivas hasta su cierre<br>Objetivo: Gestionar las acciones correctivas hasta su cierre, derivadas de un rendimiento o resultado con una desviación relevante a la planeada.<br>Practica Especifica: **SP 2.3 Gestionar las acciones correctivas**<br>Subpracticas<br>1. Gestionar las acciones hasta que finalicen<br>2. Analizar los resultados para determinar su eficacia<br>3. Determinar las acciones necesarias para corregir las desviaciones producidas a causa de la implementación de las acciones correctivas. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | No basta con aplicar las acciones correctivas a las cuestiones más importantes, es necesario gestionarlas hasta su cierre, documentando los resultados y que sirvan de referencia a tratamientos futuros. | | |

Cuadro 34. Practica 21 CMMI Nivel 2

| **Ítem** | PRA-CMM-21 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Planificación del proyecto<br>Propósito: Gestionar planes para la definición de las actividades del proyecto<br>Meta Especifica: SG 2 Desarrollar un plan de proyecto<br>Objetivo: Definir planes para gestionar y controlar la ejecución del proyecto<br>Practica Especifica: **SP 2.2 Identificar los riesgos del proyecto**<br>Subpracticas<br>1. Identificar los riesgos del proyecto, se analizan todos los peligros, amenazas y vulnerabilidades que pudieran impactar de forma negativa en la ejecución del proyecto<br>2. Documentar los riesgos<br>3. Definir acuerdos sobre lo completos y exactos de los riesgos documentados entre todas las partes.<br>4. Actualizar la información referente a los riesgos según sea necesario | | |

Cuadro 34. (Continuación)

| | |
|---|---|
| **Fase Ciclo SW** | Diseño |
| **Justificación** | De una efectiva definición de los posibles riesgos que pudiesen afectar la ejecución del proyecto se podrán obtener y definir las correspondientes acciones correctivas que de forma efectiva mitiguen estos riesgos durante las demás fases del ciclo de desarrollo de software. |

Cuadro 35. Practica 22 CMMI Nivel 2

| **Ítem** | PRA-CMM-22 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Aseguramiento de la calidad del proceso y del producto<br>Propósito: Otorgar una visión objetiva de los procesos y productos de trabajo<br>Meta Especifica: SG 1 Evaluar objetivamente los procesos y los productos de trabajo<br>Objetivo: Evaluar objetivamente los procesos y productos a los procedimientos y estándares aplicables<br>Practica Especifica: **SP 1.1 Evaluar objetivamente los procesos**<br>Subpracticas<br>1. Fomentar ambientes de trabajo entre el personal para la identificación y análisis de temas de calidad<br>2. Definir criterios para realizar las evaluaciones<br>3. Utilizar los criterios definidos para evaluar la afinidad de los procesos realizados y seleccionados con las descripciones de los estándares y procedimientos.<br>4. Identificar las no conformidades<br>5. Identificar las lecciones comprendidas que podrían optimizar los procesos. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Realizar una evaluación objetiva de la calidad de los procesos definidos em el proyecto facilitara la identificación y mejoramiento de posibles vulnerabilidades | | |

Cuadro 36. Practica 23 CMMI Nivel 2

| **Ítem** | PRA-CMM-23 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Aseguramiento de la calidad del proceso y del producto<br>Propósito: Otorgar una visión objetiva de los procesos y productos de trabajo<br>Meta Especifica: SG 1 Evaluar objetivamente los procesos y los productos de trabajo<br>Objetivo: Evaluar objetivamente los procesos y productos a los procedimientos y estándares aplicables<br>Practica Especifica: **SP 1.2 Evaluar objetivamente los productos de trabajo**<br>Subpracticas<br>1. Escoger los productos de trabajo que serán evaluados<br>2. Definir criterios para realizar las evaluaciones<br>3. Utilizar los criterios definidos para evaluar los productos de trabajo seleccionados<br>4. Realizar la evaluación de los productos de trabajo en los momentos definidos<br>5. Identificar las no conformidades<br>6. Identificar las lecciones comprendidas que podrían optimizar los procesos. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Realizar una evaluación objetiva de la calidad de los productos definidos em el proyecto facilitara la identificación y mejoramientos de posibles vulnerabilidades. | | |

Cuadro 37. Practica 24 CMMI Nivel 2

| **Ítem** | PRA-CMM-24 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Aseguramiento de la calidad del proceso y del producto<br>Propósito: Otorgar una visión objetiva de los procesos y productos de trabajo<br>Meta Especifica: SG 2 Proporcionar una visión objetiva<br>Objetivo: Hacer seguimiento y control a las no conformidades, comunicándolas de manera oportuna y asegurando su resolución<br>Practica Especifica: **SP 2.1 Comunicar y resolver las no conformidades** | | |

Cuadro 37. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas<br>1. Resolver todas las no conformidades con el personal adecuado.<br>2. Documentar las no conformidades que no pudieron ser resueltas<br>3. Escalar las no conformidades para ser tratadas en otro nivel gerencial<br>4. Analizar las no conformidades para identificar tendencias de calidad que deban identificarse y tratarse<br>5. Mantener al tanto del proceso a todas las partes interesadas.<br>6. Revisar de forma periódica las no conformidades abiertas y las tendencias con la gerencia para ser tratadas<br>7. Hacer seguimiento a las no conformidades hasta que sean resueltas. |
| **Fase Ciclo SW** | Todas las fases |
| **Justificación** | Todas aquellas no conformidades identificadas representan un riesgo para la seguridad y calidad del proyecto, por tanto, deben ser identificadas y tratadas a tiempo. |

Cuadro 38. Practica 25 CMMI Nivel 2

| **Ítem** | PRA-CMM-25 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Aseguramiento de la calidad del proceso y del producto<br>Propósito: Otorgar una visión objetiva de los procesos y productos de trabajo<br>Meta Especifica: SG 2 Proporcionar una visión objetiva<br>Objetivo: Hacer seguimiento y control a las no conformidades, comunicándolas de manera oportuna y asegurando su resolución<br>Practica Especifica: **SP 2.2 Establecer los registros**<br>Subpracticas<br>1. Guardar el registro de todas aquellas actividades realizadas para garantizar el aseguramiento de la calidad del proceso y del producto de forma detallada.<br>2. Actualizar el estado e historial de todas las actividades de aseguramiento de la calidad, cada vez que sea necesario. | | |
| **Fase Ciclo SW** | Todas las fases | | |
| **Justificación** | Documentar y registras todas aquellas acciones para garantizar el aseguramiento de la calidad permite el tratamiento adecuado de otras no conformidades. | | |

Cuadro 39. Practica 26 CMMI Nivel 2

| **Ítem** | PRA-CMM-26 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Requisitos<br>Propósito: Gestionar los requisitos de los productos, los componentes y los servicios del proyecto, garantizando la alineación ente los requisitos y los planes de trabajo del proyecto<br>Meta Especifica: SG 1 Gestionar los requisitos<br>Objetivo: Gestionar los requisitos conservando las relaciones y la alineación entre los requisitos, los planes y los productos del proyecto, implementando las acciones correctivas necesarias.<br>Practica Especifica: **SP 1.1 Comprender los requisitos**<br>Subpracticas<br>1. Definir criterios para establecer los proveedores de requisitos más adecuados.<br>2. Definir criterios para la evaluación y aceptación de requisitos<br>3. Analizar los requisitos para determinar si cumplen con los criterios definidos<br>4. Lograr la comprensión y aceptación de los requisitos entre los proveedores (Clientes) y los participantes del proyecto, para alcanzar el compromiso de las partes. | | |
| **Fase Ciclo SW** | Toma de requerimientos | | |
| **Justificación** | Antes de iniciar cualquier desarrollo, el éxito del mismo y la aceptación del producto por parte del cliente depende de una adecuada compresión de lo que quiere el proveedor y de que se debe hacer para desarrollarlo, con el objetivo de evitar reprocesos e implementar funcionalidades no requeridas | | |

Cuadro 40. Practica 27 CMMI Nivel 2

| **Ítem** | PRA-CMM-27 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Requisitos<br>Propósito: Gestionar los requisitos de los productos, los componentes y los servicios del proyecto, garantizando la alineación ente los requisitos y los planes de trabajo del proyecto<br>Meta Especifica: SG 1 Gestionar los requisitos<br>Objetivo: Gestionar los requisitos conservando las relaciones y la alineación entre los requisitos, los planes y los productos del proyecto, implementando las acciones correctivas necesarias.<br>Practica Especifica: **SP 1.2 Obtener el compromiso sobre los requisitos** | | |

Cuadro 40. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas<br>1. Evaluar el impacto de los requisitos sobre los compromisos ya existentes cuando haya un cambio o al crear un nuevo requisito.<br>2. Conciliar y dejar documentados los compromisos. |
| **Fase Ciclo SW** | Toma de requerimientos |
| **Justificación** | Ante cualquier cambio o actualización de un requerimiento es necesario negociar y lograr la comprensión y aceptación de todas las partes para establecer compromisos. |

Cuadro 41. Practica 28 CMMI Nivel 2

| **Ítem** | PRA-CMM-28 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Requisitos<br>Propósito: Gestionar los requisitos de los productos, los componentes y los servicios del proyecto, garantizando la alineación ente los requisitos y los planes de trabajo del proyecto<br>Meta Especifica: SG 1 Gestionar los requisitos<br>Objetivo: Gestionar los requisitos conservando las relaciones y la alineación entre los requisitos, los planes y los productos del proyecto, implementando las acciones correctivas necesarias.<br>Practica Especifica: **SP 1.3 Gestionar los cambios a los requisitos**<br>Subpracticas<br>1. Documentar los requisitos y sus respectivos cambios<br>2. Conservar el historial de cambios, incluyendo el análisis realizado por el cual se efectuaron.<br>3. Evaluar el impacto de los cambios a los requisitos.<br>4. Disponer de la información de los requisitos y los cambios en el proyecto | | |
| **Fase Ciclo SW** | Toma de requerimientos | | |
| **Justificación** | Los cambios efectuados a los requisitos deben estar debidamente documentados para poder estimar su volatilidad o probabilidad de que ocurran | | |

Cuadro 42. Practica 29 CMMI Nivel 2

| **Ítem** | PRA-CMM-29 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Requisitos<br>Propósito: Gestionar los requisitos de los productos, los componentes y los servicios del proyecto, garantizando la alineación ente los requisitos y los planes de trabajo del proyecto<br>Meta Especifica: SG 1 Gestionar los requisitos<br>Objetivo: Gestionar los requisitos conservando las relaciones y la alineación entre los requisitos, los planes y los productos del proyecto, implementando las acciones correctivas necesarias.<br>Practica Especifica: **SP 1.4 Mantener la trazabilidad bidireccional de los requisitos**<br>Subpracticas<br>1. Mantener la trazabilidad de los requisitos<br>2. Mantener la trazabilidad desde los requisitos principales y sus derivados hasta la asignación de los productos de trabajo<br>3. Crear una matriz de trazabilidad | | |
| **Fase Ciclo SW** | Toma de requerimientos | | |
| **Justificación** | Mantener la trazabilidad de los requisitos desde el nivel más bajo y simple hasta el nivel más complejo, permite detectar con facilidad si el producto y componentes del proyecto fueron desarrollados conforme fueron concebidos. | | |

Cuadro 43. Practica 30 CMMI Nivel 2

| **Ítem** | PRA-CMM-30 | **Metodología** | CMMI Nivel 2 |
|---|---|---|---|
| **Descripción** | Área de proceso: Gestión de Requisitos<br>Propósito: Gestionar los requisitos de los productos, los componentes y los servicios del proyecto, garantizando la alineación ente los requisitos y los planes de trabajo del proyecto<br>Meta Especifica: SG 1 Gestionar los requisitos<br>Objetivo: Gestionar los requisitos conservando las relaciones y la alineación entre los requisitos, los planes y los productos del proyecto, implementando las acciones correctivas necesarias.<br>Practica Especifica: **SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos** | | |

Cuadro 43. (Continuación)

| | |
|---|---|
| **Descripción** | Subpracticas<br>1. Revisar que los planes y actividades del proyecto sean consistentes con los requisitos y cambios en los mismos.<br>2. Identificar inconsistencias y su origen<br>3. Identificar los cambios que deberían realizarse en los planes y líneas de trabajo derivados de los cambios en los requisitos<br>4. Realizar las acciones correctivas necesarias |
| **Fase Ciclo SW** | Toma de requerimientos |
| **Justificación** | Garantizar que el desarrollo del proyecto esté acorde a los lineamientos establecidos en los requisitos. |

### 4.1.3 PERSONAL SOFTWARE PROCESS (PSP)

A continuación, se describen las practicas seleccionadas en el estándar PSP que tienen alguna relación o pueden aplicarse al desarrollo de software, indicando la práctica identificada, además se conserva el orden en que son tratadas en la guía oficial de la metodología.

Cuadro 44. Practica 1 PSP

| **Ítem** | PRA-PSP-1 | **Metodología** | PSP |
|---|---|---|---|
| **Descripción** | Revisión personal del código fuente para encontrar defectos | | |
| **Fase Ciclo SW** | Codificación | | |
| **Justificación** | El factor más importante en la calidad de un programa es el compromiso personal, la revisión del código personal es el principal método para la eliminación de defectos. Para un ingeniero de software es esencial aprender a gestionar todos los defectos que introducen en sus programas; entender la clase de defectos que se pueden introducir, reunir datos de defectos y crear un perfil de defectos personales (lista de comprobación personal). Una forma de hacer más eficiente la práctica es revisar el histórico de defectos. | | |

Cuadro 45. Practica 2 PSP

| **Ítem** | PRA-PSP-2 | **Metodología** | PSP |
|---|---|---|---|
| **Descripción** | Utilización de datos de los defectos | | |
| **Fase Ciclo SW** | Planificación | | |
| **Justificación** | A la hora de producir programas de alta calidad se deben emplear medidas consistentes, utilizar los datos de los defectos permite determinar cómo mejorar la prevención o localización de los defectos que se hayan introducido en el código.<br>Utilizando la lista de comprobación personal para la revisión del código y con la ayuda de histórico de defectos y el tamaño del nuevo programa a codificar, se puede estimar el número de defectos que se introducirán, creando la planificación de defectos que se incluye en el plan del proyecto para obtener una mejor precisión en la etapa de planificación y por ende una mejor calidad y seguridad en el producto final. | | |

Cuadro 46. Practica 3 PSP

| **Ítem** | PRA-PSP-3 | **Metodología** | PSP |
|---|---|---|---|
| **Descripción** | Revisión del diseño | | |
| **Fase Ciclo SW** | Diseño | | |
| **Justificación** | Una de las prácticas que se utiliza en la metodología es la revisión de defectos en el diseño, aunque si bien los defectos más comunes se presentan en la fase de codificación, las funciones de lógica, rendimiento y sincronización son defectos propios del diseño.<br>Es importante hacer el diseño lógico y funcional en la fase de diseño y no durante la codificación, esta recomendación brindada por PSP es debido a que la mayoría de ingenieros trabajan el diseño sobre la marcha. Las representaciones de diseño precisas pueden ahorrar tiempo de implementación y reducir los defectos de diseño. Una pobre representación puede causar defectos e inseguridades en el producto; las representaciones más comunes son las gráficas, el seudocódigo y las matemáticas. | | |

### 4.1.4 TEAM SOFTWARE PROCESS (TSP)

A continuación, se describen las practicas seleccionadas en el estándar TSP que tienen alguna relación o pueden aplicarse al desarrollo de software, indicando la práctica identificada, además se conserva el orden en que son tratadas en la guía oficial de la metodología.

Cuadro 47. Practica 1 TSP

| **Ítem** | PRA-TSP-1 | **Metodología** | TSP |
|---|---|---|---|
| **Descripción** | Plan de calidad | | |
| **Fase Ciclo SW** | Despliegue | | |
| **Justificación** | Durante el lanzamiento del equipo TSP, los ingenieros realizan un plan de calidad donde estiman a partir de datos históricos el porcentaje de errores que se inyectan en cada fase del desarrollo de software y de igual forma la estimación de como eliminarlos. Una vez establecido el plan de calidad el equipo lo revisa para verificar que los parámetros de calidad sean razonables y validar el cumplimiento de metas y objetivos de calidad del producto. | | |

Cuadro 48. Practica 2 TSP

| **Ítem** | PRA-TSP-2 | **Metodología** | TSP |
|---|---|---|---|
| **Descripción** | Perfil de calidad | | |
| **Fase Ciclo SW** | Despliegue y mantenimiento | | |
| **Justificación** | El perfil de calidad mide los datos de proceso de un módulo Contra los estándares de calidad de la organización, a través de cinco dimensiones (datos para el diseño, revisiones de diseño, revisión de código, defectos de compilación, defectos de prueba de unidad) los ingenieros pueden revisar e identificar productos con problemas de calidad. | | |

## 4.2 CATEGORIZAR LAS BUENAS PRACTICAS EN SEGURIDAD DE DESARROLLO

A continuación, se describe la identificación y generación de criterios o principios de seguridad que permitan categorizar cada una de las practicas seleccionadas con base en los estándares y metodologías estudiados en el presente documento, para este proceso se tomó como referencia la guía generada por el colectivo internacional Information Systems Security Association[30] y su proyecto GAIPS (Generrally Accepted Information Security Principles) que busca la unificación de principios de seguridad de TI que han sido aprobados y aceptados por profesionales en el área de seguridad de la comunidad en general y a los cuales muchas otras organizaciones internacionales se han sumado para ser parte del proyecto. Con el objetivo de darle un enfoque más práctico y de dirigir estos principios a la presente investigación dada su naturaleza solo se toman algunos conceptos presentados en la guía de GAIPS como referencia y se adaptaron a la investigación para determinar cuáles prácticas de las seleccionadas proveen una mejor implementación de seguridad en el desarrollo de software.

- ✓ Criterio: CRI 1
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Trazabilidad
  - Descripción: Conservar la trazabilidad de los cambios efectuados en los procesos, componentes e información.

    La implementación de prácticas con este componente permite administrar y gestionar de forma efectiva las versiones o cambios presentados en los productos y configuraciones, facilitando el proceso de identificación y control de incidentes sin mayores repercusiones para todo el proceso.

---

[30] INFORMATION SYSTEMS SECURITY ASSOCIATION. Generrally Accepted Information Security Principles, Version 3.0 [En linea]. [citado 29 junio de 2018]. Disponible en: <http://all.net/books/standards/GAISP-v30.pdf>

- ✓ Criterio: CRI 2
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Disponibilidad
  - Descripción: Gestionar de forma efectiva los recursos disponibles.
    La implementación de prácticas con este componente permite hacer uso de los recursos de forma eficiente y segura para evitar sobrecargas que produzcan algún incidente de seguridad.

- ✓ Criterio: CRI 3
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Trazabilidad
  - Descripción: Estimar el nivel de asimilación y afinidad de los procesos realizados durante la ejecución frente a lo planificado.
    La implementación de prácticas con este componente permite medir durante la realización del proyecto el cumplimiento de los requerimientos y lineamientos establecidos en la planeación y diseño, identificando la necesidad de aplicar acciones correctivas de ser necesario.

- ✓ Criterio: CRI 4
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Integridad, Confiabilidad
  - Descripción: Identificar y controlar las fallas de forma oportuna.
    La implementación de prácticas con este componente permite facilitar el proceso de identificación y tratamiento de fallos de forma oportuna, reduciendo los tiempos de respuesta.

- ✓ Criterio: CRI 5
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Autenticidad
  - Descripción: Asignar y verificar los derechos de accesibilidad de los usuarios. La implementación de prácticas con este componente permite validar la autenticidad de los usuarios en el sistema.

- ✓ Criterio: CRI 6
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Disponibilidad
  - Descripción: Definir elementos de configuración para la ejecución de procesos y procedimientos.
    La implementación de prácticas con este componente permite establecer los elementos de configuración necesarios que serán el incentivo para el desarrollo del proyecto

- ✓ Criterio: CRI 7
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Integridad, Confiabilidad, Trazabilidad
  - Descripción: Gestionar y evaluar los productos y procesos.
    La implementación de prácticas con este componente permite evaluar los productos y procesos que requieren atención, identificando el incumplimiento de requisitos de seguridad establecidos para proceder a mejorarlos.

- ✓ Criterio: CRI 8
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Integridad
  - Descripción: Comprender y aceptar los requisitos de implementación.

La implementación de prácticas con este componente permite una adecuada comprensión de los requisitos del proyecto a desarrollar, además de la aceptación y compromiso de las partes interesadas para su implementación.

- ✓ Criterio: CRI 9
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Confiabilidad
  - Descripción: Emplear mecanismos para proteger el sistema de ataques internos o externos.

    La implementación de prácticas con este componente permite implementar mecanismos o herramientas que faciliten la protección del sistema o aplicación contra ataques de origen interno o externo.

- ✓ Criterio: CRI 10
  - Categoría: Funcionalidad
  - Dimensión de seguridad: Integridad, Confiabilidad, Trazabilidad
  - Descripción: Ejecutar procedimientos de evaluación sobre la funcionalidad y aceptación del sistema.

    La implementación prácticas con este componente permite evaluar el grado de cumplimiento y aceptación del sistema una vez finalizada la fase de desarrollo.

- ✓ Criterio: CRI 11
  - Categoría: Información
  - Dimensión de seguridad: Integridad
  - Descripción: Asegurar la autenticidad y valides de la información.

    La implementación de prácticas con este componente permite asegurar que la información se conserve tal cual como fue concebida y no sea manipulada.

- ✓ Criterio: CRI 12
  - Categoría: Información
  - Dimensión de seguridad: Confiabilidad
  - Descripción: Definir las necesidades de información y categorizarla según su importancia.

    La implementación de prácticas con este componente permite determinar que necesidades de información surgen durante la ejecución del proyecto, priorizarlas según su importancia y establecer su tratamiento de forma segura.

- ✓ Criterio: CRI 13
  - Categoría: Información
  - Dimensión de seguridad: Integridad, trazabilidad
  - Descripción: Administrar y almacenar de forma segura la información.

    La implementación prácticas con este componente permite la gestión segura de la información durante las transacciones que impliquen algún movimiento de información.

- ✓ Criterio: CRI 14
  - Categoría: Información
  - Dimensión de seguridad: Confiabilidad
  - Descripción: Conservar la confidencialidad de la información.

    La implementación prácticas con este componente Permite definir parámetros y procedimientos seguros para salvaguardar la información de accesos o divulgación no autorizados.

- ✓ Criterio: CRI 15
  - Categoría: Recurso Humano
  - Dimensión de seguridad: Integridad, Confiabilidad

- Descripción: Establecer compromisos y responsabilidades de los actores del proceso.

  La implementación de prácticas con este componente permite definir las acciones propuestas para las partes implicadas en el desarrollo del proyecto de software.

Con base en la definición de los criterios establecidos en el paso anterior se llevó a cabo la construcción de la matriz de evaluación cualitativa que relaciona cada practica identificada de los estándares y metodologías estudiados con los respectivos factores de seguridad más destacados para el tratamiento y almacenamiento de información acordes con cada una de las fases del ciclo de vida del software, generando una ponderación para cada una de ellas según los criterios definidos en la tabla de calificación descrita a continuación:

Cuadro 49. Valores de clasificación de criterios

| Valor | Clasificación | Descripción |
|---|---|---|
| 1 | Bajo | Bajo grado de afinidad y cumplimiento del criterio |
| 2 | Medio | Mediano grado de afinidad y cumplimiento del criterio |
| 3 | Alto | Alto grado de afinidad y cumplimiento del criterio |

Se procede a evaluar cada practica seleccionada según los criterios establecidos y la clasificación definida de la siguiente manera:

Cuadro 50. Evaluación de prácticas de seguridad en el desarrollo de software

| Matriz de Criterios de Evaluación | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Practicas\Criterios** | **Ciclo de SW** | | | | | | | **CRI1** | **CRI2** | **CRI3** | **CRI4** | **CRI5** | **CRI6** | **CRI7** | **CRI8** | **CRI9** | **CRI10** | **CRI11** | **CRI12** | **CRI13** | **CRI14** | **CRI15** | **TOTAL** |
| | **Planeación** | **Requerimient** | **Diseño** | **Desarrollo** | **Pruebas** | **Despliegue** | **Mantenimient** | | | | | | | | | | | | | | | | | |
| PRA-ISO-1 | | | | x | | | | 3 | 3 | 3 | 3 | 1 | 1 | 3 | 1 | 2 | 1 | 3 | 2 | 3 | 2 | 2 | 33 |
| PRA-ISO-2 | x | x | x | x | x | x | x | 3 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 1 | 3 | 2 | 1 | 26 |
| PRA-ISO-3 | | x | x | | | | | 1 | 3 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 22 |
| PRA-ISO-4 | | | | x | x | x | x | 2 | 1 | 3 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 | 2 | 1 | 2 | 2 | 33 |
| PRA-ISO-5 | | | | x | | | | 1 | 2 | 1 | 3 | 1 | 2 | 3 | 1 | 3 | 2 | 3 | 2 | 2 | 2 | 1 | 29 |
| PRA-ISO-6 | | | x | x | | | | 3 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 26 |
| PRA-ISO-7 | | | x | | | | | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 27 |
| PRA-ISO-8 | | x | | | | | | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 1 | 3 | 2 | 3 | 2 | 2 | 29 |
| PRA-ISO-9 | | | | x | x | | | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 1 | 28 |
| PRA-ISO-10 | | | | x | | | | 3 | 2 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 1 | 36 |
| PRA-ISO-11 | | x | x | | | | | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 3 | 2 | 1 | 3 | 2 | 2 | 3 | 1 | 28 |
| PRA-ISO-12 | | | | x | x | | | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 1 | 28 |
| PRA-ISO-13 | | | | | x | | | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 2 | 1 | 26 |
| PRA-CMMI-1 | x | x | x | x | x | x | x | 1 | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 20 |
| PRA-CMMI-2 | x | x | x | x | x | x | x | 2 | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 26 |
| PRA-CMMI-3 | x | x | x | x | x | x | x | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 20 |
| PRA-CMMI-4 | x | x | x | x | x | x | x | 3 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 22 |

Cuadro 50. (Continuación)

| Matriz de Criterios de Evaluación | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Practicas\Criterios | Ciclo de SW | | | | | | | CRI1 | CRI2 | CRI3 | CRI4 | CRI5 | CRI6 | CRI7 | CRI8 | CRI9 | CRI10 | CRI11 | CRI12 | CRI13 | CRI14 | CRI15 | TOTAL |
| | Planeación | Requerimient | Diseño | Desarrollo | Pruebas | Despliegue | Mantenimient | | | | | | | | | | | | | | | | | |
| PRA-CMMI-5 | x | x | x | x | x | x | x | 3 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 3 | 2 | 1 | 1 | 24 |
| PRA-CMMI-6 | x | x | x | x | x | x | x | 3 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 22 |
| PRA-CMMI-7 | x | x | x | x | x | x | x | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 22 |
| PRA-CMMI-8 | x | x | x | x | x | x | x | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 18 |
| PRA-CMMI-9 | x | x | x | x | x | x | x | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 3 | 2 | 2 | 1 | 2 | 23 |
| PRA-CMMI-10 | x | x | x | x | x | x | x | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 20 |
| PRA-CMMI-11 | x | x | x | x | x | x | x | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 1 | 27 |
| PRA-CMMI-12 | x | x | x | x | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 16 |
| PRA-CMMI-13 | x | x | x | x | x | x | x | 2 | 3 | 3 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 27 |
| PRA-CMMI-14 | x | x | x | x | x | x | x | 2 | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 3 | 23 |
| PRA-CMMI-15 | x | x | x | x | x | x | x | 2 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 21 |
| PRA-CMMI-16 | x | x | x | x | x | x | x | 3 | 2 | 3 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 27 |
| PRA-CMMI-17 | x | x | x | x | x | x | x | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 3 | 27 |
| PRA-CMMI-18 | x | x | x | x | x | x | x | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 20 |
| PRA-CMMI-19 | x | x | x | x | x | x | x | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 3 | 22 |
| PRA-CMMI-20 | x | x | x | x | x | x | x | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 19 |
| PRA-CMMI-21 | x | | | | | | | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 21 |

Cuadro 50. (Continuación)

| Matriz de Criterios de Evaluación | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Practicas\Criterios | Ciclo de SW | | | | | | | CRI1 | CRI2 | CRI3 | CRI4 | CRI5 | CRI6 | CRI7 | CRI8 | CRI9 | CRI10 | CRI11 | CRI12 | CRI13 | CRI14 | CRI15 | TOTAL |
| | Planeación | Requerimie | Diseño | Desarrollo | Pruebas | Despliegue | Mantenimie | | | | | | | | | | | | | | | | |
| PRA-CMMI-22 | x | x | x | x | x | x | x | 2 | 1 | 3 | 2 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 22 |
| PRA-CMMI-23 | x | x | x | x | x | x | x | 2 | 1 | 3 | 2 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 22 |
| PRA-CMMI-24 | x | x | x | x | x | x | x | 3 | 2 | 1 | 3 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 3 | 25 |
| PRA-CMMI-25 | x | x | x | x | x | x | x | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 17 |
| PRA-CMMI-26 | | | x | | | | | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 1 | 3 | 23 |
| PRA-CMMI-27 | | | x | | | | | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 20 |
| PRA-CMMI-28 | | | x | | | | | 3 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 24 |
| PRA-CMMI-29 | | | x | | | | | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 22 |
| PRA-CMMI-30 | | | x | | | | | 2 | 2 | 3 | 3 | 1 | 2 | 2 | 3 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 29 |
| PRA-PSP-1 | | | | x | | | | 1 | 2 | 2 | 3 | 1 | 2 | 3 | 2 | 2 | 2 | 3 | 2 | 1 | 2 | 1 | 29 |
| PRA-PSP-2 | x | | x | | | | | 2 | 2 | 1 | 3 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 25 |
| PRA-PSP-3 | | | x | | | | | 1 | 1 | 1 | 3 | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 22 |
| PRA-TSP-1 | | | | | | x | | 2 | 3 | 1 | 2 | 1 | 2 | 3 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 25 |
| PRA-TSP-2 | | | | | | x | x | 3 | 2 | 2 | 2 | 1 | 2 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 1 | 1 | 27 |

Luego de realizar la evaluación de cada una de las practicas seleccionadas se obtuvo como resultado que las prácticas que cumplían con un mayor número de criterios y superaban la media del total cuantificado fueron las siguientes:

- PRA-ISO-1: Gestión del Cambio
- PRA-ISO-2: Separación de las instalaciones de desarrollo, ensayo y operación.
- PRA-ISO-4: Aceptación del sistema.
- PRA-ISO-5: Controles contra códigos maliciosos.
- PRA-ISO-6: Registro de Usuarios.
- PRA-ISO-7: Gestión de privilegios.
- PRA-ISO-8: Análisis y especificación de los requisitos de seguridad
- PRA-ISO-9: Validación de los datos de entrada.
- PRA-ISO-10: Control de procesamiento interno.
- PRA-ISO-11: Integridad del mensaje.
- PRA-ISO-12: Validación de los datos de salida.
- PRA-ISO-13: Protección de los datos de prueba del sistema.
- PRA-CMMI-2: Establecer un sistema de gestión de configuración
- PRA-CMMI-11: Almacenar los datos y los resultados
- PRA-CMMI-13: Monitorizar los parámetros de planificación del proyecto
- PRA-CMMI-16: Llevar a cabo las revisiones del progreso
- PRA-CMMI-17: Llevar a cabo las revisiones de hitos
- PRA-CMMI-30: Asegurar el alineamiento entre el trabajo del proyecto y los requisitos
- PRA-PSP-1: Revisión personal del código fuente para encontrar defectos
- PRA-TSP-2: Perfil de calidad

En resumen, en total fueron seleccionadas veinte (20) prácticas que según la cuantificación dada cumplen en su totalidad o en gran medida con los criterios de evaluación de seguridad definidos y que formaran parte de la guía de buenas prácticas de desarrollo de software formulada en el presente documento.

## 4.3 FACTORES DE RIESGO ASOCIADOS A LAS PRÁCTICAS SELECCIONADAS

Para la definición de los riesgos asociados se utilizó como documento de referencia la publicación "Pautas de la evaluación del riesgo de la seguridad" disponible en el portal web del Estado de Massachusetts[31] y la norma ISO/IEC 27005[32] en los cuales se establecen las pautas para el tratamiento de riesgos informáticos, cabe aclarar que en el presente análisis se requiere la identificación de factores de riesgo de prácticas seguras y no sobre activos de información por tanto estos textos de referencia solo se usaron como pautas o guías según la relación al área de acción de cada práctica, por tanto a continuación se presentan las descripciones propuestas para identificar de forma general el posible factor de riesgo asociado a la práctica seleccionada.

PRA-ISO-1: Gestión del Cambio

- El exceso de documentación y de cambios no controlados pueden dificultar en gran medida su administración y la evaluación (auditoria) de los procesos afectados, lo que puede representar un riesgo ya que se podrían pasar por alto impactos no calculados producto de un cambio.

---

[31] ESTADO DE MASSACHUSETTS, USA. Information Security Risk Assessment Guidelines [En linea]. MASS. [citado 29 junio de 2018]. Disponible en: <http://www.mass.gov/anf/research-and-tech/cyber-security/security-for-state-employees/risk-assessment/risk-assessment-guideline.html>

[32] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology – Security techniques - Information security risk management. ISO IEC 27005. Mahdi:ISO. 2011.

PRA-ISO-2: Separación de las instalaciones de desarrollo, ensayo y operación.

- El hecho de tener entornos separados de desarrollo implica disponer de una configuración uniforme y segura para cada uno de los ambientes de trabajo que se tienen, el más ligero cambio o diferencia podría generar riesgos para el correcto funcionamiento del sistema.

PRA-ISO-4: Aceptación del sistema.

- Durante el ciclo de vida de un software es necesario aplicar las pruebas suficientes que garanticen el correcto funcionamiento del sistema y cumplan con los requerimientos y compromisos establecidos entre todos las partes, con el objeto de lograr cumplir con los criterios de aceptación definidos, dado que el hecho de hacer más o menos de lo que se había pedido, puede implicar reprocesos riesgosos para la organización.

PRA-ISO-5: Controles contra códigos maliciosos.

- Son tantos y tan diversos los ataques por códigos maliciosos que muchas veces por tratar de salvaguardar los sistemas de información se recurren a configuración excesivas o poco fiables que pueden no estar cumpliendo su trabajo y por el contrario representar nuevas vulnerabilidades.

PRA-ISO-6: Registro de Usuarios.

- Es necesario definir procedimientos formales para ingresar y dar de baja a los usuarios en el sistema, para evitar que queden usuarios activos que ya no lo son y represente una puerta de acceso a personal no autorizado.

PRA-ISO-7: Gestión de privilegios.

- Disponer de una adecuada administración de privilegios en donde solo el personal autorizado pueda acceder a las funciones que le fueron designadas y no ocurran modificaciones no programadas o escalamiento de privilegios.

PRA-ISO-8: Análisis y especificación de los requisitos de seguridad

- Al especificar los requerimientos de seguridad deben ser tenidos en cuenta las condiciones del entorno, funcionalidad y concurrencia del sistema desarrollo, para tener en consideración todos aquellos factores de riesgo que amenazan el funcionamiento del sistema.

PRA-ISO-9: Validación de los datos de entrada.

- Una validación incorrecta o incompleta de los datos de entrada en el sistema pueden representar una vulnerabilidad para la ejecución de scripts y códigos maliciosos.

PRA-ISO-10: Control de procesamiento interno.

- No validar de forma continua la información procesada dentro del sistema puede abrir la puerta a errores de procesamiento o de acciones malintencionadas que pueden provocar un mal funcionamiento del sistema e incluso la interrupción del servicio.

PRA-ISO-11: Integridad del mensaje.

- La ausencia de validación de la autenticidad e integridad de un mensaje, puede repercutir en la interceptación de los datos trasmitidos lo que podría comprometer la seguridad de la información.

PRA-ISO-12: Validación de los datos de salida.

- Una validación incorrecta o incompleta de los datos de salida en el sistema pueden representar una vulnerabilidad ya que se puede generar información errada o confidencial sin percibirse el error.

PRA-ISO-13: Protección de los datos de prueba del sistema.

- El no proteger de forma correcta los datos de prueba podría exponer de manera indiscriminada información sensible para el cliente.

PRA-CMMI-2: Establecer un sistema de gestión de configuración

- Todos los parámetros de configuración deben ser previamente establecidos así mismo los elementos necesarios para la ejecución de los procesos, el no tener procedimientos formales podría implicar ambigüedades o fallos del sistema que podrían ser explotados.

PRA-CMMI-11: Almacenar los datos y los resultados

- No conservar los datos históricos de medición y acciones correctivas implementadas durante el proceso de desarrollo, podrían representar reprocesos sobre problemas ya solucionados para futuros desarrollos

PRA-CMMI-13: Monitorizar los parámetros de planificación del proyecto

- Es necesario controlar todos los aspectos de ejecución del proyecto, para evitar la aparición de no conformidades o incidentes no deseados por el cumplimiento de los compromisos adquiridos en la planeación y los requerimientos.

PRA-CMMI-16: Llevar a cabo las revisiones del progreso

- No estimar y revisar continuamente el avance y ejecución del proyecto de desarrollo en cada etapa puede representar reprocesos o retrasos costosos.

PRA-CMMI-17: Llevar a cabo las revisiones de hitos

- El hecho de no tratar una cuestión o problema no identificado por falencias en la revisión podría representar errores de procesamiento o vulnerabilidades que podrían ser explotadas.

PRA-CMMI-30: Asegurar el alineamiento entre el trabajo del proyecto y los requisitos

- Cualquier desviación del desarrollo frente a los requisitos del sistema y de seguridad especificados, podrían llevar al incumplimiento y no aceptación del sistema por parte del cliente, lo que puede llevar a reprocesos.

PRA-PSP-1: Revisión personal del código fuente para encontrar defectos

- Una revisión inadecuada del código, podría representar que el sistema se despliegue con errores o vulnerabilidades que en un ambiente de producción pueden ser explotados.

PRA-TSP-2: Perfil de calidad

- Un proceso inadecuado de medición de la calidad de un módulo especifico puede permitir que pasan errores como si todo estuviera bien, representando un riesgo para el sistema.

## 4.4 GUÍA DE BUENAS PRÁCTICAS DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE

El resultado más importante del presente proyecto es la “GUÍA DE BUENAS PRÁCTICAS DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE CON BASE EN ESTÁNDARES RECONOCIDOS EN EMPRESAS DE DESARROLLO DE SOFTWARE”. Con el propósito de facilitar su utilización, la guía se presenta en un documento independiente, con su propia diagramación y numeración. El documento se adjunta en un archivo con el mismo nombre.

# 5 RESULTADOS Y DISCUSION

Luego del análisis realizado a los controles y herramientas descritos por cada uno de los estándares que han sido objeto de estudio en el presente documento y teniendo en cuenta que son metodologías reconocidas a nivel internacional en el tema de seguridad y calidad, mas no específicamente enfocadas al desarrollo de software seguro como fue el motivo de la investigación se identificaron un total de 48 practicas o controles segregadas de la siguiente manera de la ISO 27001 13 practicas, CMMI nivel 2 30 practicas, PSP 3 prácticas y TSP 2 practicas.

Al obtener esos resultados se pensó que la mejor manera de evaluar estas prácticas fue seleccionar un conjunto de criterios que si bien no cubrieron la totalidad de aspectos que se pudiesen presentar a la hora de desarrollar un software que se pueda considerar seguro, por lo menos si integraron en gran medida los paradigmas de seguridad que pueden atender a buen número de las amenazas circundantes en los sistemas de información de la actualidad, producto de esta evaluación se obtuvo como resultado un conjunto de 20 buenas prácticas de seguridad aplicables al desarrollo de software, en donde para cada una de las fases del ciclo de vida se asociaron algunas de estas prácticas.

Si bien cada uno de los estándares y metodologías seleccionados en este estudio tienen enfoques muy diversos en cuanta la forma de asumir el paradigma de seguridad, además de que algunos de ellos parten de una premisa de calidad, pero que de una u otra manera derivan en este paradigma y dando por hecho que muchos otros fueron concebidos para la fundamentación de todo el proyecto de software o de la misma infraestructura de TI y no tanto para el desarrollo, es necesario denotar y de paso justificar que es ahí donde el concepto base de control de seguridad tiene mayor aplicabilidad porque el objetivo era emplear estas prácticas desde el momento justo en que se concibe la idea de misma sistema hasta

cuándo es generado el producto final de una forma estandarizada y que durante todo el proceso de desarrollo se entendiera que el componente de seguridad es una parte fundamental del mismo y no algo que se debe tener en cuenta al final, lo que puede producir una respuesta más eficiente ante los posibles riesgos y amenazas que se pudiesen presentar.

# 6 CONCLUSIONES

- En la actualidad existen múltiples metodologías de seguridad en el desarrollo de software disponibles en el mercado que proporcionan resultados muy diversos durante su implementación lo que genera en su gran mayoría resultados positivos para la protección y manejo de la información, cada una de estas metodologías tiene características positivas y negativas y son aplicables en diferentes entornos, la gran mayoría de desarrolladores tiene muy buenos conocimientos en el desarrollo de software y están provistos de múltiples habilidades, pero muchos de ellos no siguen un proceso formalizado o simplemente hacen uso de prácticas que les facilitan el proceso de desarrollo, pero que no son realmente seguras y no cumplen con los lineamientos y normas de seguridad existentes.

- Si los procesos de desarrollo de software no se ejecutan eficientemente garantizando la seguridad y calidad de los mismos podría generar grandes riesgos para la información almacenada en esos sistemas desarrollados, el hecho de contemplar los controles de la norma ISO 27001 que rigen sobre el desarrollo seguro y el acceso a los sistemas de información facilitaría el gran medida la protección de los datos, además de dar cumplimiento a estándares internacionales en seguridad de tecnologías de la información.

- Garantizar la seguridad en el desarrollo de software no solo se basa en el uso y aplicación de unas metodologías, normas, disposiciones legales o conjunto de prácticas durante el desarrollo, sino que se puede complementar a través de la definición formal de un proceso en donde se estandarice la forma en que se debe realizar la implementación del software con miras a disminuir el error mediante la optimización y calidad del producto.

- Por consiguiente la elaboración de una guía de buenas prácticas de desarrollo de software que contenga las actividades más destacadas descritas por una metodología segura de desarrollo como PSP/TSP, bajo un conjunto de prácticas para la estandarización del ciclo de vida del software como lo expone CCMI nivel dos de madurez, todo esto enmarcado en los controles establecidos por la norma ISO 27001 concernientes al desarrollo de software facilitarían el proceso de implementación de nuevas soluciones de software optimizando el recurso, garantizando la calidad y resguardando efectivamente la información.

- La implementación de una guía de buenas prácticas de seguridad, basada en estándares reconocidos, en el ciclo de vida de desarrollo de software mejoraría la confiabilidad y calidad del producto generado, crearía una experiencia satisfactoria para el desarrollador, en donde mediante el uso de metodologías y prácticas bien definidas podrá usar eficientemente los recursos a su disposición, mitigando considerablemente los riesgos presentes durante todas las fases del desarrollo, además representaría mayor confianza y satisfacción de parte del cliente en el producto generado y por consiguiente en la empresa.

# 7 RECOMENDACIONES

El uso de la guía de buenas prácticas desarrollada permite blindar en un grado considerable el proceso de confección de un producto de software y a su vez hacerlo de forma estandarizada, pero es necesario recordar que es prácticamente inconcebible lograr dar una cobertura y tratamiento total a todas las amenazas existente, por tanto, esta guía se convierte en una buena herramienta que ayudara a mejorar la calidad y seguridad en la implementación de un sistema de información mas no es una solución definitiva.

# BIBLIOGRAFÍA


ACAR, Yasemin; STRANSKY, Christian; WERMKE, Dominik; WEIR, Charles; MAZUREK, Michelle L. y FAHL. Sascha. Developers Need Support, Too: A Survey of Security Advice for Software Developers [En linea]. Leibniz University Hannover, CISPA, Saarland University, Security Lancaster, University of Maryland. IEEE. 2017. pp. 22 – 26.

AGENCIA DE CALIDAD SANITARIA DE ANDALUCÍA, Consejería de Salud. Estrategia de calidad y seguridad en aplicaciones móviles de salud [En Línea]. Andalucía, España, 2012. [citado 29 junio de 2018]. Disponible en: <http://www.calidadappsalud.com>

ALWAN, Motea. What is System Development Life Cycle? [En linea]. Airbrake. Enero, 2015. [citado 29 junio de 2018]. Disponible en: <https://airbrake.io/blog/sdlc/what-is-system-development-life-cycle>

ÁREA DE AUDITORÍA Y CONTROL. COBIT: Modelo para auditoría y control de sistemas de información [En linea]. Universidad EAFIT, Boletín 54. Mayo, 2007. [citado 29 junio de 2018]. Disponible en: <http://www.eafit.edu.co/escuelas/administracion/consultorio-contable/Documents/boletines/auditoria-control/b13.pdf>

BRITO ABUNDIS, Carlos Joaquín. Metodologías para desarrollar software seguro. [En línea]. Universidad Autónoma de Zacatecas. Diciembre de 2013. [citado 29 junio de 2018]. Disponible en: <http://recibe.cucei.udg.mx/revista/es/vol2-no3/pdf/computacion05.pdf>

DIRECCIÓN NACIONAL DE PROTECCIÓN DE DATOS PERSONALES. Guía de buenas prácticas para el desarrollo de Apps [En línea]. Ministerio de Justicia y Derechos Humanos, Argentina, abr. 2015 [citado 29 junio de 2018]. Disponible en: <http://www.jus.gob.ar/media/3075908/guiabpsoftware.pdf>

DUSSAN CLAVIJO, Ciro Antonio. Políticas de seguridad informática. ENTRAMADO [En línea]. Junio de 2016. [citado 29 junio de 2018]. Disponible en: <http://www.unilibrecali.edu.co/images2/revista-entramado/pdf/pdf_articulos/volumen2/Politicas_de_seguridad_informtica.pdf>

ESTADO DE MASSACHUSETTS, USA. Information Security Risk Assessment Guidelines [En linea]. MASS. [citado 29 junio de 2018]. Disponible en: <http://www.mass.gov/anf/research-and-tech/cyber-security/security-for-state-employees/risk-assessment/risk-assessment-guideline.html>

GASCA HURTADO, Gloria P. Análisis de riesgos para el desarrollo de software seguro. Universidad Politécnica de Madrid [En línea]. Agosto de 2006. [citado 29 junio de 2018]. Disponible en: <http://www.dlsiis.fi.upm.es/docto_lsiis/Trabajos20052006/Gasca.pdf>

GARCIA, I; ANDREA, I. Using the Software Process Improvement approach for Defining a Methodology for Embedded Systems Development using the CMMI-DEV v1.2 [En linea]. Technological University of the Mixtec. Mexico. IEEE. 2010. pp. 233–240.

GILLIAM D.P.. Security risks: management and mitigation in the software life cycle [En linea]. Jet Propulsion Laboratory. California Inst. of Technology. Pasadena, California. IEEE. 2005. pp. 319 – 325.

HILBURN, T.B. y HUMPHREY, W.S. Teaching teamwork in Software [En línea]. IEEE. 2012. pp. 72 – 77.

HUTTER, David. Physical Security and Why It Is Important [En linea]. SANS Institute InfoSec Reading Room. Junio, 2016. [citado 29 junio de 2018]. Disponible en: <https://www.sans.org/reading-room/whitepapers/physical/physical-security-important-37120>

INFORMATION SYSTEMS SECURITY ASSOCIATION. Generrally Accepted Information Security Principles, Version 3.0 [En linea]. [citado 29 junio de 2018]. Disponible en: <http://all.net/books/standards/GAISP-v30.pdf>

INSTITUTO COLOMBIANO DE NORMAS TECNICAS Y CERTIFICACION. Técnicas de seguridad. Sistemas de Gestión de la Seguridad de la Información (SGSI). NTC-ISO IEC 2700. Bogotá: ICONTEC. Marzo, 2006.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology – Security techniques - Information security risk management. ISO IEC 27005. Mahdi:ISO. 2011.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology, Security techniques, Evaluation criteria for IT security, Part 1: Introduction and general model ISO. ISO/IEC 15408-1:2005. Suiza: ISO. 1999.

JBW GROUP INTERNATIONAL INFORMATION ASSURENCE. Evolution of an International Information Security Standard [En linea]. JBWGroup. Abril, 2010. [citado 29 junio de 2018]. Disponible en: <http://www.jbwgroup.com/assets/PDFs/JBW%20Group%20-%20EU%20-%20InfoSec%20History%20V2-N2.1.pdf>

JIMÉNEZ ROJAS, Jesús Ramon. La seguridad informática y el usuario final [En linea]. Revista Digital Universitaria, UNAM. Abril, 2018. [citado 29 junio de 2018]. Disponible en: <http://www.revista.unam.mx/vol.9/num4/art20/art20.pdf>

KHARI, Manju. Embedding Security in Software Development Life Cycle (SDLC) [En linea]. Dept. of Computer Science IEEE. 2016. pp. 2182 – 2186.

LÓPEZ NEIRA, A. y RUIZ SPOHR, J. El portal de ISO 27001 en español [En línea]. [citado en citado 29 junio de 2018]. Disponible en: < http://www.iso27000.es/>

LOPEZ PROVENCIO, Ferran. Metodologías para el desarrollo de software seguro [En Linea]. Enero de 2015. [citado 29 junio de 2018]. Disponible en: <https://upcommons.upc.edu/bitstream/handle/2099.1/24902/103275.pdf>

MARULANDA L. César, CEBALLOS H. Julián. UNA REVISIÓN DE METODOLOGÍAS SEGURAS EN CADA FASE DEL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE [En línea]. Enero de 2012. [citado 29 junio de 2018]. Disponible en: <http://web.usbmed.edu.co/usbmed/fing/v3n1/v3n1a2.pdf>

MCGRAW, G. Software Security [En línea]. CIGITAL. Marzo, 2004. [citado 29 junio de 2018]. Disponible en: <https://www.cigital.com/papers/download/software-security-gem.pdf>

MINISTERIO DEL INTERIOR. Políticas De Seguridad De La Información - Seguridad Física [En linea]. Julio, 2014. [citado 29 junio de 2018]. Disponible en: <http://www.mininterior.gov.co/sites/default/files/oip-2014-psi-especificas-4_seguridad_fisica.doc>

PAULK, Mark. C. A History of the Capability Maturity Model for Software [En Linea]. Carnegie Mellon University. 2009. [citado 29 junio de 2018]. Disponible en:

<https://pdfs.semanticscholar.org/6fb0/c324e08698a9e364693151605a74982b487a.pdf>

PORT, Daniel; KAZMAN, Rick; TAKENAKA, Ann. Strategic Planning for Information Security and Assurance [En linea]. Department of Information Technology Management, University of Hawaii. IEEE. 2008. pp. 466-471.

RAMINGWONG, Sakgasit; RAMINGWONG, Lachana. Implementing a Personal Software Process (PSP SM). Course: A Case Study [En linea]. Department of Computer Engineering, Faculty of Engineering, Chiang Mai University, Thailand. Abril, 2012. [citado 29 junio de 2018]. Disponible en: <URL: http://dx.doi.org/10.4236/jsea.2012.58074>

RAMÍREZ BRAVO, Pia. y DONOSO JAURES, Felipe. METODOLOGÍA ITIL: Descripción, Funcionamiento y Aplicaciones [En linea]. Facultad de Ciencias Económicas y Administrativas, Universidad De Chile. [citado 29 junio de 2018]. Disponible en: <http://repositorio.uchile.cl/tesis/uchile/2006/donoso_f/sources/donoso_f.pdf>

ROUSE, Margaret. Information security (infosec) [En linea]. Searchsecurity. Techtarget. Septiembre, 2016. [citado 29 junio de 2018]. Disponible en: <https://searchsecurity.techtarget.com/definition/information-security-infosec>

SOFTWARE ENGENIEERING INSTITUTE. CMMI® para Desarrollo, Versión 1.3 [En línea]. Noviembre 2010. [citado 29 junio de 2018]. Disponible en: <https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_28782.pdf>

SOFTWARE ENGENIEERING INSTITUTE. The Personal Software ProcessSM (PSPSM) Body of Knowledge, Version 2.0 [En linea]. Agosto 2009. [citado 29 junio de 2018]. Disponible en:

<https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13751.pdf>

SOFTWARE ENGENIEERING INSTITUTE. The Team Software Process SM (TSP SM) [En linea]. SEI. Noviembre, 2000. [citado 29 junio de 2018]. Disponible en: <https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13754.pdf>

SORIANO, Miguel. Seguridad en redes y seguridad de la información [En línea]. Improvet. Republica Checa. Primera Edición. 2014. [citado 29 junio de 2018]. Disponible en:<http://improvet.cvut.cz/project/download/C2ES/Seguridad_de_Red_e_Informacion.pdf>

UNIVERSIDAD INTERNACIONAL DE VALENCIA. Conceptos sobre seguridad lógica informática [En linea]. VIU. Marzo, 2018. [citado 29 junio de 2018]. Disponible en: <https://www.universidadviu.com/conceptos-seguridad-logica-informatica/>

WANG, Wenjin. Development of Mass Spectrometer Software Project Based on CMMI [En Linea]. Purification Equipment Research Institute of CSIC. IEEE. 2017. pp. 2508 – 2511.

WATTS S. Humphrey. Noviembre 2000. The Personal Software Process SM (PSPSM) [En linea]. [citado 29 junio de 2018]. Disponible en: <https://resources.sei.cmu.edu/asset_files/SpecialReport/2009_003_001_15029.pdf>

WHITMORE, Jim; TOBIN, Will. Improving Attention to Security in Software Design with Analytics and Cognitive Techniques [En linea]. USA. IEEE. 2017. pp. 16 – 21.

# GUÍA DE BUENAS PRÁCTICAS DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE CON BASE EN ESTÁNDARES RECONOCIDOS EN EMPRESAS DE DESARROLLO DE SOFTWARE

Especialización en Seguridad Informática

**Camilo Fernández Bernal** ▸ Universidad Nacional Abierta y a Distancia (UNAD) ▸ 07/09/2018

# ÍNDICE

Buenas prácticas para proteger el entorno de desarrollo

## INTRODUCCION

El creciente uso de las tecnologías de la información y la comunicación ha denotado también un incremento exponencial del número de amenazas existentes en la red, lo que ha generado la necesidad de desarrollar sistemas de información y aplicaciones más seguras que garanticen la integridad, confidencialidad y disponibilidad de los datos.

En la actualidad la gran mayoría de empresas han optado por caracterizar y sistematizar toda su información almacenándola en sistemas que faciliten su administración y estén acordes con las exigencias del mercado. No es para menos la preocupación de las organizaciones por salvaguardar sus datos, ya que hoy en día la información se ha constituido como el activo más importante que poseen y ante tantas amenazas es necesario protegerse de la mejor manera.

Para las empresas de desarrollo software y desarrolladores independientes la seguridad es en una herramienta que mejora la calidad de sus productos y los protege ante cualquier vulnerabilidad que se pudiese presentar. Por consiguiente, ya no solo basta con desarrollar software por desarrollar, es necesario que se haga uso de buenas prácticas y metodologías de seguridad en el desarrollo de software que faciliten la creación de sistemas confiables que estén acordes con las exigencias del mercado y que incluyan el componente de seguridad en cada proceso y modulo implementado.

El presente documento pretende generar una guía de buenas prácticas de seguridad para el desarrollo de software basadas en primera instancia en una metodología de desarrollo seguro como lo es PSP/TSP, en estándares internacionales de seguridad como lo sugiere la norma ISO 27001 y en la estandarización del proceso de desarrollo de las empresas mediante las buenas prácticas que implementa el nivel 2 de madurez de CMMI.

## BUENAS PRACTICAS DE SEGURIDAD

A continuación, se describen las practicas ordenas según las fases del ciclo de vida del desarrollo de software a la cual aplican, iniciando con las prácticas que son transversales a todo el proceso, es decir que son aplicables en todas las fases:

### Practica 1: Monitorizar los parámetros de planificación del proyecto

**Descripción**

1. Gestionar el avance del proyecto frente a lo establecido en el calendario.

2. Gestionar el costo y esfuerzo real empleado frente al planeado y determinar las desviaciones significativas.

3. Gestionar los parámetros o atributos de cada producto para determinar su desarrollo e identificar las desviaciones significativas frente a lo planeado.

4. Gestionar los recursos

5. Gestionar la capacidad del personal del proyecto

6. Documentar las desviaciones significativas.

**Fase ciclo SW:**

Todas las fases

**Implementación:**

Con miras a la optimización de recursos y al cumplimiento de los objetivos propuesto en el proyecto de desarrollo, es necesario monitorear cada aspecto y parámetro del mismo, y compáralo con lo presupuestado inicialmente, de tal forma que este se ajuste a lo que se había planeado y de haber algún desvió significativo se puedan aplicar las respectivas acciones correctivas.

**Factor de Riesgo:**

Es necesario controlar todos los aspectos de ejecución del proyecto, para evitar la aparición de no conformidades o incidentes no se deseados por el cumplimiento de los compromisos adquiridos en la planeación y los requerimientos.

## Practica 2: Asegurar el alineamiento entre el trabajo del proyecto y los requisitos

**Descripción**

1. Revisar que los planes y actividades del proyecto sean consistentes con los requisitos y cambios en los mismos.

2. Identificar inconsistencias y su origen

3. Identificar los cambios que deberían realizarse en los planes y líneas de trabajo derivados de los cambios en los requisitos

4. Realizar las acciones correctivas necesarias

**Fase ciclo SW:**

Todas las fases

**Implementación:**

Garantizar que el desarrollo del proyecto esté acorde a los lineamientos establecidos en los requisitos.

**Factor de Riesgo:**

Cualquier desviación del desarrollo frente a los requisitos del sistema y de seguridad especificados, podrían llevar al incumplimiento y no aceptación del sistema por parte del cliente, lo que puede llevar a reprocesos.

## Practica 3: Llevar a cabo las revisiones del progreso

**Descripción**

1. Comunicar con frecuencia el estado de las actividades y productos del proyecto a las partes interesadas.

2. Revisar y analizar las mediciones realizadas

3. Identificar y documentar las desviaciones significativas frente a lo planeado

4. Documentar las solicitudes de cambio para los productos y procesos

5. Documentar los resultados

6. Gestionar las peticiones de cambio hasta su cierre.

**Fase ciclo SW:**

Todas las fases

**Implementación:**

Aplicar mediciones a los productos y procesos del proyecto en diferentes momentos de su desarrollo, permite controlar el estado, calidad y rendimiento del mismo.

**Factor de Riesgo:**

No estimar y revisar continuamente el avance y ejecución del proyecto de desarrollo en cada etapa puede representar reprocesos o retrasos costosos.

## Practica 4: Llevar a cabo las revisiones de hitos

**Descripción**

1. Revisar los hitos más significativos con las partes interesadas en determinados momentos del calendario.

2. Realizar una revisión de compromisos, estados, planes y riesgos del desarrollo del proyecto.

3. Identificar los temas más relevantes y sus respectivos impactos

4. Documentar los resultados

5. Gestionar las acciones hasta su cierre.

**Fase ciclo SW:**

Todas las fases

**Implementación:**

Revisar los eventos más destacadas durante todo el desarrollo del proyecto por todas las partes interesadas, permitirá hacer seguimiento del estado del mismo y facilitando el proceso de toma de decisiones para implementar las acciones necesarias para la obtención de los resultados esperados.

**Factor de Riesgo:**

El hecho de no tratar una cuestión o problema no identificado por falencias en la revisión podría representar errores de procesamiento o vulnerabilidades que podrían ser explotadas.

## Practica 5: Almacenar los datos y los resultados

**Descripción**

1. Revisar la integridad, precisión y actualización de la información

2. Almacenar la información de las mediciones y análisis conforme el procedimiento general de almacenamiento de datos.

3. Restringir el acceso a los datos almacenados solo a personal autorizado.

4. Gestionar y dar buen uso de la información almacenada

**Fase ciclo SW:**

Todas las fases

**Implementación:**

El almacenamiento de todos los datos históricos de la medición servirá de base para fundamentar la implementación de criterios y mediciones futuras, además también permite mejorar el proceso de ejecución del proyecto basado en experiencias pasadas.

**Factor de Riesgo:**

No conservar los datos históricos de medición y acciones correctivas implementadas durante el proceso de desarrollo, podrían representar reprocesos sobre problemas ya solucionados para futuros desarrollos

## Practica 6: Separación de las instalaciones de desarrollo, ensayo y operación

**Descripción**

Evidenciar si la organización cuenta con entornos separados de desarrollo, pruebas y producción, conservando la similitud de configuración y recursos para que se trabaje de forma uniforme en cada entorno. Con el objetivo de disminuir cambios no deseados en el entorno de producción.

**Fase ciclo SW:**

Todas las fases

**Implementación:**

En el ciclo de vida del software es necesario disponer de entornos de desarrollo separados que permite realizar las pruebas de funcionamiento correspondientes a nuevas implementaciones sin afectar el funcionamiento de las existentes y reduciendo el riesgo de modificaciones o cambios inesperados durante todas las fases.

**Factor de Riesgo:**

El hecho de tener entornos separados de desarrollo implica disponer de una configuración uniforme y segura para cada uno de los ambientes de trabajo que se tienen, el más ligero cambio o diferencia podría generar riesgos para el correcto funcionamiento del sistema.

## Practica 7: Establecer un sistema de gestión de configuración

**Descripción**

1. Definir metodología para administrar diferentes niveles de gestión

2. Proporcionar controles de acceso autorizado al sistema

3. Almacenar y recuperar los elementos de configuración en el sistema

4. Compartir los elementos de configuración identificados entre los niveles de gestión

5. Almacenar y recuperar los registros de gestión

6. Generar informes de gestión de configuración desde el sistema de gestión

7. Preservar la información del sistema de gestión

8. Realizar las modificaciones necesarias a la estructura de gestión.

**Fase ciclo SW:**

Todas las fases

**Implementación:**

Durante todo el ciclo de desarrollo de software para los elementos que requieren una configuración es necesario generar un sistema de gestión que administre de forma efectiva y eficiente la configuración de los mismos durante el desarrollo.

**Factor de Riesgo:**

Todos los parámetros de configuración deben ser previamente establecidos así mismo los elementos necesarios para la ejecución de los procesos, el no tener procedimientos formales podría implicar ambigüedades o fallos del sistema que podrían ser explotados.

## Practica 8: Análisis y especificación de los requisitos de seguridad

**Descripción**

Para toda nueva implementación de nuevos sistemas o actualización de alguno existente se deben especificar los requerimientos de seguridad necesarios.

**Fase ciclo SW:**

Toma de requerimientos

**Implementación:**

Durante el proceso de toma y especificación de requerimientos es indispensable para un desarrollo seguro establecer los requerimientos de seguridad necesarios para proteger la integridad del software y la información, garantizando un adecuado funcionamiento ante la presencia del algún incidente de seguridad, además deben estar estrechamente relacionados con las políticas de seguridad del cliente.

**Factor de Riesgo:**

Al especificar los requerimientos de seguridad deben ser tenidos en cuenta las condiciones del entorno, funcionalidad y concurrencia del sistema desarrollo, para tener en consideración todos aquellos factores de riesgo que amenazan el funcionamiento del sistema.

## Practica 9: Integridad del mensaje

**Descripción**

Identificar y establecer los requisitos necesarios para garantizar la veracidad e integridad de los mensajes en la aplicación, implementando los controles necesarios.

**Fase ciclo SW:**

Toma de requerimientos

**Implementación:**

Durante la fase de toma de requerimientos y de diseño se deben identificar qué clase de controles o métodos se implementarán para las transferencias de información, por ejemplo, encriptando los mensajes. Estos controles a implementar durante el desarrollo deben poder garantizar que la información enviada y recibida se conserva tal cual como fue concebida y sea recibida por su destinatario inicial.

**Factor de Riesgo:**

La ausencia de validación de la autenticidad e integridad de un mensaje, puede repercutir en la interceptación de los datos trasmitidos lo que podría comprometer la seguridad de la información.

### Practica 10: Gestión de privilegios

**Descripción**

Se deben controlar y gestionar la asignación y uso de los privilegios de usuarios en el sistema.

**Fase ciclo SW:**

Diseño

**Implementación:**

Durante el diseño se deben identificar y especificar de forma clara todos los actores que intervendrán en el software y que solo puedan desempeñar el rol para el cual fue asignado, restringiendo cualquier otra acción no permitida.

**Factor de Riesgo:**

Disponer de una adecuada administración de privilegios en donde solo el personal autorizado pueda acceder a las funciones que le fueron designadas y no ocurran modificaciones no programadas o escalamiento de privilegios.

## Practica 11: Registro de Usuarios

**Descripción**

Se deben establecer procedimientos formales para el registro y retiro de usuarios en los sistemas de información.

**Fase ciclo SW:**

Diseño, desarrollo

**Implementación:**

Durante el desarrollo es necesario considerar una adecuada administración de usuarios que permita el acceso solo a las personas registradas en el sistema.

**Factor de Riesgo:**

Es necesario definir procedimientos formales para ingresar y dar de baja a los usuarios en el sistema, para evitar que queden usuarios activos que ya no lo son y represente una puerta de acceso a personal no autorizado.

## Practica 12: Controles contra códigos maliciosos

**Descripción**

Establecimiento de controles para la identificación y prevención de ataques por código malicioso y procedimientos para su recuperación.

**Fase ciclo SW:**

Desarrollo

**Implementación:**

Durante el desarrollo es necesario definir y codificar procedimientos para la prevención de ataques por códigos maliciosos, de tal forma que la información que será ingresada al sistema sea filtrada previamente antes de ser almacenada, en especial cuando refiere a consultas en la base de datos.

**Factor de Riesgo:**

Son tantos y tan diversos los ataques por códigos maliciosos que muchas veces por tratar de salvaguardar los sistemas de información se recurren a configuración excesivas o poco fiables que pueden no estar cumpliendo su trabajo y por el contrario representar nuevas vulnerabilidades.

## Practica 12: Gestión del Cambio

**Descripción**

Evidenciar si se tiene control de los cambios efectuados en los sistemas de procesamiento de información conservando y administrando log de cambios.

**Fase ciclo SW:**

Desarrollo

**Implementación:**

Durante el desarrollo de software se deben identificar los elementos que deben estar bajo control para restringir y documentar los cambios realizados sobre estos, a su vez dichos cambios o versiones del desarrollo deben ser auditados o revisados para garantizar su completitud e integridad.

**Factor de Riesgo:**

El exceso de documentación y de cambios no controlados pueden dificultar en gran medida su administración y la evaluación (auditoria) de los procesos afectados, lo que puede representar un riesgo ya que se podrían pasar por alto impactos no calculados producto de un cambio.

## Practica 14: Control de procesamiento interno

**Descripción**

Definir controles de validación para identificar cualquier corrupción de la información derivada de fallos en su procesamiento o acciones malintencionadas.

**Fase ciclo SW:**

Desarrollo

**Implementación:**

Es necesario validar que la información que se encontrara almacenada en el aplicativo se conserve integra y corresponda a la ingresada en forma y contenido.

**Factor de Riesgo:**

No validar de forma continua la información procesada dentro del sistema puede abrir la puerta a errores de procesamiento o de acciones malintencionadas que pueden provocar un mal funcionamiento del sistema e incluso la interrupción del servicio.

## Practica 15: Revisión personal del código fuente para encontrar defectos

**Descripción**

El factor más importante en la calidad de un programa es el compromiso personal, la revisión del código personal es el principal método para la eliminación de defectos. Para un ingeniero de software es esencial aprender a gestionar todos los defectos que introducen en sus programas; entender la clase de defectos que se pueden introducir, reunir datos de defectos y crear un perfil de defectos personales (lista de comprobación personal). Una forma de hacer más eficiente la práctica es revisar el histórico de defectos.

**Fase ciclo SW:**

Desarrollo

**Implementación:**

La revisión personal del código fuente permite una mayor exploración y entendimiento de lo que fue desarrollo, además de que facilita la identificación de defectos. Con la práctica constante y el historial de defectos encontrados se puede llegar generar patrones de comportamiento para fortalecer falencias y así generar código seguro y de calidad.

**Factor de Riesgo:**

Una revisión inadecuada del código, podría representar que el sistema se despliegue con errores o vulnerabilidades que en un ambiente de producción pueden ser explotados.

### Practica 16: Validación de los datos de entrada

**Descripción**

Validación de la correctitud e integridad de la información ingresada en la aplicación.

**Fase ciclo SW:**

Desarrollo, pruebas

**Implementación:**

Durante el desarrollo es necesario establecer controles para filtrar y controlar la información que ingresa al aplicativo de forma que solo ingrese información valida, descartando toda información errada o que contenga códigos maliciosos.

**Factor de Riesgo:**

Una validación incorrecta o incompleta de los datos de entrada en el sistema pueden representar una vulnerabilidad para la ejecución de scripts y códigos maliciosos.

### Practica 17: Validación de los datos de salida

**Descripción**

Se deben validar los datos de salida de la aplicación, para garantizar que el procesamiento de la información almacenada en la aplicación sea correcto y que sea la espera.

**Fase ciclo SW:**

Desarrollo, pruebas

**Implementación:**

Los datos que ingresan y son procesados por la aplicación deben ser controlados y revisados antes de salir de la aplicación, para ello es necesario que el sistema tenga controles de la validación que permitan constatar la consistencia y veracidad de los datos generados.

**Factor de Riesgo:**

Una validación incorrecta o incompleta de los datos de entrada en el sistema pueden representar una vulnerabilidad ya que se puede generar información errada o confidencial sin percibirse el error.

## Practica 18: Aceptación del sistema

**Descripción**

Se deben establecer criterios de aceptación de nuevas implementaciones o actualización de las existentes, además de realizar las respectivas pruebas de funcionamiento durante su desarrollo, previo a su aprobación.

**Fase ciclo SW:**

Desarrollo, pruebas, despliegue y mantenimiento

**Implementación:**

Durante el desarrollo es necesario aplicar las pruebas suficientes que garanticen el correcto funcionamiento del sistema en cada fase, con el objetivo de lograr cumplir con los criterios de aceptación y poder pasar a la etapa de producción.

**Factor de Riesgo:**

Durante el ciclo de vida de un software es necesario aplicar las pruebas suficientes que garanticen el correcto funcionamiento del sistema y cumplan con los requerimientos y compromisos establecidos entre todos las partes, con el objeto de lograr cumplir con los criterios de aceptación definidos, dado que el hecho de hacer más o menos de lo que se había pedido, puede implicar reprocesos riesgosos para la organización.

## Practica 19: Protección de los datos de prueba del sistema

**Descripción**

Los datos de prueba se deben pasar por un proceso idóneo de selección según las especificaciones del sistema, estos datos deben ser controlados y resguardados.

**Fase ciclo SW:**

Pruebas

**Implementación:**

En la fase de pruebas un componente decisivo para garantizar el funcionamiento correcto y esperado del sistema depende directamente de los datos de prueba seleccionados, estos deben guardar concordancia con la información real que será ingresada y debe considerar múltiples escenarios de uso que garanticen la integridad de la información procesada.

**Factor de Riesgo:**

El no proteger de forma correcta los datos de prueba podría exponer de manera indiscriminada información sensible para el cliente.

## Practica 20: Perfil de calidad

**Descripción**

El perfil de calidad mide los datos de proceso de un módulo contra los estándares de calidad de la organización, a través de cinco dimensiones (datos para el diseño, revisiones de diseño, revisión de código, defectos de compilación, defectos de prueba de unidad) los ingenieros pueden revisar e identificar productos con problemas de calidad.

**Fase ciclo SW:**

Despliegue, mantenimiento

**Implementación:**

La generación de perfiles de calidad basados en los estándares definidos por la organización, permitirá caracterizar y evaluar la calidad del producto en base a la obtención de esas metas propuestas en el proyecto y desarrolladas por el equipo de desarrollo.

**Factor de Riesgo:**

Un proceso inadecuado de medición de la calidad de un módulo especifico puede permitir que pasan errores como si todo estuviera bien, representando un riesgo para el sistema.

## CONCLUSIONES

En la actualidad existen múltiples metodologías de seguridad en el desarrollo de software disponibles en el mercado que proporcionan resultados muy diversos durante su implementación lo que genera en su gran mayoría resultados positivos para la protección y manejo de la información, cada una de estas metodologías tiene características positivas y negativas y son aplicables en diferentes entornos, la gran mayoría de desarrolladores tiene muy buenos conocimientos en el desarrollo de software y están provistos de múltiples habilidades, pero muchos de ellos no siguen un proceso formalizado o simplemente hacen uso de prácticas que les facilitan el proceso de desarrollo, pero que no son realmente seguras y no cumplen con los lineamientos y normas de seguridad existentes.

Si los procesos de desarrollo de software no se ejecutan eficientemente garantizando la seguridad y calidad de los mismos podría generar grandes riesgos para la información almacenada en esos sistemas desarrollados, el hecho de contemplar los controles de la norma ISO 27001 que rigen sobre el desarrollo seguro y el acceso a los sistemas de información facilitaría el gran medida la protección de los datos, además de dar cumplimiento a estándares internacionales en seguridad de tecnologías de la información.

Garantizar la seguridad en el desarrollo de software no solo se basa en el uso y aplicación de una metodologías, normas, disposiciones legales o conjunto de prácticas durante el desarrollo, sino que se puede complementar  a través de la definición formal de un proceso en donde se estandarice la forma en que se debe realizar la implementación del software con miras a disminuir el error mediante la optimización y calidad del producto como lo describe algunas de las practicas descritas por CMMI en su nivel dos de madurez.

La implementación de una guía de buenas prácticas de seguridad, basada en estándares reconocidos, en el ciclo de vida de desarrollo de software mejoraría la confiabilidad y calidad del producto generado, crearía una experiencia satisfactoria para el desarrollador, en donde mediante el uso de metodologías y prácticas bien definidas podrá usar eficientemente los recursos a su disposición, mitigando considerablemente los riesgo presentes durante todas las fases del desarrollo, además representaría mayor confianza y satisfacción de parte del cliente en el producto generado y por consiguiente en la empresa.

## MAS INFORMACION

[1] Software Engenieering Institute. CMMI® para Desarrollo, Versión 1.3.
http://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf

[2] El portal de ISO 27001 en español
http://www.iso27000.es/

[3] CMMI Institute
http://cmmiinstitute.com/get-started

[4] WATTS S. Humphrey. The Personal Software ProcessSM (PSPSM)
http://www.sei.cmu.edu/reports/00tr022.pdf

[5] Software Engenieering Institute. TSP and CMMI: A Brief History, Versión 1.3.
https://www.sei.cmu.edu/tsp/tsp-history.cfm

[6] Software Engenieering Institute. Team Software ProcessSM (TSPSM) Body of Knowledge (BOK)
http://www.sei.cmu.edu/reports/10tr020.pdf