UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor:

Bach. Díaz Castañeda Rudy Slaytonw

https://orcid.org/0000-0002-2057-1244

Asesor:

Mg. Mejía Cabrera Heber Iván

https://orcid.org/0000-0002-0007-0928

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú

2021

APROBACIÓN DEL JURADO

MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

_____

Bach. Díaz Castañeda Rudy Slaytonw
Autor

_____

Mg. Mejía Cabrera Heber Iván
Asesor

_____ _____

Mg. Bravo Ruiz Jaime Arturo
Presidente de Jurado

Mg. Bances Saavedra David Enrique
Secretario de Jurado

_____

Mg. Tuesta Monteza Victor Alexci
Vocal de Jurado

DEDICATORIA

Dedicado a mis padres Rudy Díaz e Isela Castañeda, por darme la educación que me permite cumplir mis sueños; a mi pareja Lucía Ventura por apoyarme y motivarme a seguir adelante con mis metas, y a todas las personas que estuvieron ahí para alentarme y apoyarme en las adversidades.

AGRADECIMIENTO

A Dios por su gracia y los dones que me ha dado; a mi asesor de tesis, Iván Heber Mejía Cabrera, por la motivación en esta investigación y ser un apoyo durante la carrera y mis amigos que fueron gran compañía durante la realización del proyecto.

RESUMEN

Actualmente, las empresas han aumentado considerablemente el uso de tecnologías, lo que supone mayores vulnerabilidades en su software utilizado, este motivo, obliga a las empresas desarrolladoras de sistemas, buscar nuevos métodos de desarrollo que apliquen seguridad a sus productos sin elevar demasiado los costos, y es ahí donde nacen los modelos de procesos orientados al desarrollo de software con características de seguridad. La presente investigación busca brindar una solución eficaz a este problema, ya que se basa en los modelos de procesos, para aplicar seguridad al software, mas importantes como son OWASP SAMM y Microsoft-SDL, los cuales se han modificado y resumido para obtener un estándar rápido de utilizar y que garantice una protección alta a las vulnerabilidades de seguridad más recurrentes. Se seleccionaron las vulnerabilidades más recurrentes basándonos en una investigación realizada por OWASP, del mismo se obtuvieron las técnicas para evitar estas vulnerabilidades, esta información fue utilizada como apoyo principal del modelo de procesos para asi poder evitar estas vulnerabilidades de manera correcta en todas las etapas del ciclo de vida del desarrollo de software.

PALABRAS CLAVE

Modelo de Procesos, Desarrollo de Software, Seguridad, Ciclo de Vida.

ABSTRACT

At present, the use of technologies in companies has increased considerably, which means greater vulnerabilities in their software used, this reason, obliges systems development companies, look for new development methods that apply security to their products without raising costs too much, and that is where the process models oriented to software development with security features are born. The present research seeks to provide an effective solution to this problem, since it is based on process models, to apply security to the software, more important such as OWASP SAMM and Microsoft-SDL, which have been modified and summarized to obtain a standard Quick to use and ensure high protection for the most recurring security vulnerabilities. The most recurrent vulnerabilities were selected based on an investigation carried out by OWASP, the same techniques were obtained to avoid these vulnerabilities, this information was used as the main support of the process model in order to avoid these vulnerabilities correctly at all stages of the Software development life cycle.

KEYWORDS

Process Model, Software Development, Security, Life Cycle

ÍNDICE

I. INTRODUCCIÓN

1.1. Planteamiento del problema

En las últimas décadas, el internet se ha convertido en algo indispensable para las empresas y mayoría de personas, y dentro de esa línea, la seguridad informática es una necesidad creciente para la protección de la información corporativa. Debido a esto, se ha despertado un gran interés en desarrollar un código seguro para que no solo existan aplicaciones de alta calidad, sino que también tengan seguridad, ya que la mayor parte de los ataques están enfocados a estas y es necesario que las empresas apliquen metodologías y herramientas para que estas aplicaciones cumplan con las exigencias debidas y así evitar este problema en el futuro.

Sin embargo, hay que tener en cuenta que la mayoría de desarrolladores no tienen la habilidad y el conocimiento necesario para realizar este tipo de trabajo, por lo que las empresas corren un gran riesgo con sus activos de información, así pues, esta investigación se propone evaluar y crear un modelo de procesos sencillo y eficiente, en base al proceso del Ciclo de vida de desarrollo de software (SDLC), el cual plantea considerar todas las etapas del proceso (Requisitos, Diseño, Desarrollo y Pruebas) como un micro proyecto, es decir, que las etapas se verán reflejadas en todo el proyecto hasta lograr el objetivo deseado, en comparación al modelo tradicional, que no tiene la seguridad implementada en sus procesos y buenas prácticas.

La sociedad continúa transformándose digitalmente más rápido que nunca, donde la tecnología de información (TI) se ha convertido en una parte esencial de una organización en su gestión y operaciones tecnológicas. Esto se demuestra en el consumo de software por parte de las empresas en el transcurso de los años, el cual ha ido aumentando en gran medida. Estudios realizados, como el Gartner Says Global IT Spending to Grow 1.1 Percent in 2019 publicado por (Gartner, 2019), muestra la cantidad de dinero asignado a servicios de TI en el año 2018 donde está incluido el software para empresas.

También se puede observar un pronóstico para el año 2019 sobre el aumento de consumo por parte de las empresas en el área de TI.

Lo más importante es hacer un diagnóstico de lo que se debe hacer para reconocer el problema en los sistemas como identificar los activos de información que se deben proteger, cómo protegerlos, cuáles son sus vulnerabilidades y cómo mitigarlas. También es necesario identificar patrones de ataque en todas las fases y almacenarlos en una base de conocimiento para así prevenir futuros ataques en otras aplicaciones.

Tabla 1.

Pronóstico mundial de gasto en TI (miles de millones de dólares estadounidenses).

	2018 Gasto	**2018 Crecimiento (%)**	**2019 Gasto**	**2019 Crecimiento (%)**
Sistemas de Centro de Datos	210	15.5	204	-2.8
Software empresarial	399	9.3	427	7.1
Dispositivos	667	0.3	655	-1.9
Servicios informáticos	982	5.5	1,016	3.5
Servicios de comunicaciones	1,489	2.1	1,487	-0.1
General IT	**3,747**	**4.0**	**3,790**	**1.1**

Fuente: (Gartner, 2019)

Las empresas desarrolladoras de software (EDS) cumplen un importante papel en esta transformación digital, siendo estas las responsables de

proveer soluciones tecnológicas a las empresas, principalmente software para gestión y operaciones.

Estas EDS realizan un proceso complejo para poder desarrollar un software específico para determinada empresa, el cual debe ser realizado de manera satisfactoria para concluir un proyecto correctamente. Existen varias formas de realizar esta tarea, pero las más habitual es usar una metodología para desarrollar el sistema, la cual es una guía para poder segmentar etapas y acciones a realizar en el SDLC. El SDLC comprende los siguientes puntos: Requisitos, Análisis, Diseño, Codificación y Prueba.

La etapa crucial es la de pruebas, porque en ella se encuentran los errores del software desarrollado, lo cual conlleva al principal problema de todo el desarrollo de software, gastos elevados en la corrección de errores, que en muchas ocasiones son la principal razón del fracaso del proyecto. Como explica (Laínez Fuentes, Noriega Martínez, Ramos, & Durango, 2015) en su estudio sobre Seguridad en el SDLC, el costo de solucionar errores aumenta cuanto más tarde se logren encontrar, lo cual sucede a menudo debido que la mayoría de procesos de desarrollo no revisan vulnerabilidades o bien lo hacen en la etapa final.

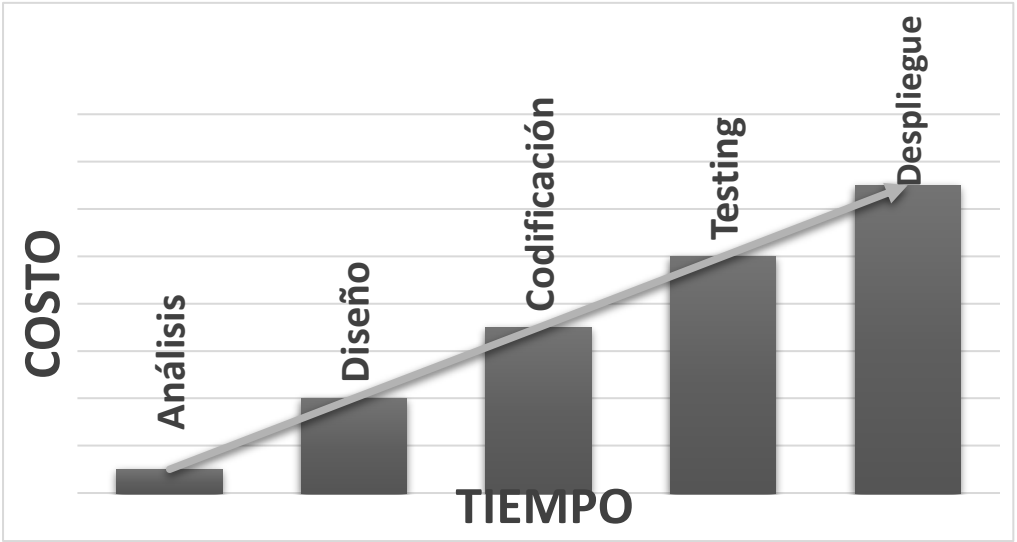


Figura 1.Costos en solucionar errores a través de las etapas de desarrollo de software. (Milano, 2007)

Entre los problemas usuales que se encuentran en esta etapa final, es la falta de seguridad en la aplicación, lo cual pone en peligro la información del cliente, este riesgo tiene que ser asumido por la EDS la cual tiene el deber

de encontrar las posibles amenazas de seguridad y reducir el riesgo lo máximo posible, pero todas estas actividades suponen un coste para la EDS y esto perjudica completamente el proyecto.

Muchos de estos problemas de seguridad son conocidos desde ya hace bastante tiempo, muchos a pesar de su antigüedad siguen afectando el software. Según el estudio realizado en OWASP top 10 - 2017 (Owasp.org., 2017), se tienen registrados las diez vulnerabilidades de seguridad de aplicaciones web críticos del 2017. De cada uno de estos se describen múltiples maneras de proteger el software, esto quiere decir que ya se tiene el conocimiento de los problemas de seguridad más recurrentes en el desarrollo de software y también el cómo evitarlos, pero se siguen cometiendo errores al momento del desarrollo debido a la falta de un modelo de procesos que especifique explícitamente que pasos se deben realizar para evitar dichos problemas.

1.2. Antecedentes de Estudio

Según el estudio “Embedding security in Software Development Life Cycle (SDLC)” (Khari, 2016), desarrollada en el Dept. of Computer Science AIACT&R, Arabia Saudita, abordó el problema de riesgo y seguridad en el SDLC, en el que toma como punto de referencia el SDLC tradicional el ámbito de la ingeniería de software, donde demostró que la seguridad se convierte en una preocupación primordial, y a pesar que varios tipos de modelos SDLC y técnicas de pruebas de seguridad estaban disponibles, pocos han hecho una integración de SDLC con los factores de seguridad. Algunos modelos definen los aspectos de seguridad en su arquitectura, pero pocos discuten modelos básicos de software. Sin embargo, las funcionalidades de riesgo de incrustación, junto con la seguridad en tradicionales SDLC ha traído una nueva definición. Por lo que el autor propone un SDLC que ha incluido el riesgo y características de seguridad. Como resultado, las ventajas de ambas técnicas se han fusionado en una sola, y tienen en cuenta la Gestión de Riesgos como otro componente importante que se aplica a todo el modelo.

Como se explica en “A Methodology for Enhancing Software Security during Development Processes,” (Shehab Farhan & Mostafa Mostafa, 2018), desarrollada en la Naif Arab University for Security Sciences, Arabia Saudita enfrentaron el problema de seguridad del software en el ciclo de desarrollo, en el cual demostraron que la falta de procesos seguros en el SDLC es crucial para este. Los autores proponen una metodología para la mejora de la seguridad del software en el ciclo de desarrollo, integrando las medidas de seguridad en todos los procesos en cada iteración de todo el SDLC en lugar de medir sólo en la fase de prueba. Además, proponen algunas medidas que se utilizan para evaluar la seguridad en cada paso del desarrollo del ciclo vital. Además, llegan a la conclusión que, siguiendo la metodología propuesta en la construcción de aplicaciones de software a través de procesos de ejecución, la aplicación sería más segura, de buena calidad y con un buen rendimiento. La metodología propuesta se implementa en un proceso de desarrollo de software real y los resultados demuestran que este método ha mejorado la seguridad del software.

Según el estudio realizado en “Reframing Security in Contemporary Software Development Life Cycle” (Frijns, Bierwolf, & Zijderhand, 2019), desarrollada en el Ministry of Interior The Hague, Australia, enfrentaron la falta de seguridad en la construcción de sistemas de información, mediante la comparación de los conceptos de Agile Scrum y DevOps, a lo largo de las fases SDLC. Los autores confirman que hay una cobertura limitada de la seguridad en esos marcos. Y que hay solamente una cobertura parcial de la seguridad en el enfoque DevOps y lo hace principalmente en las etapas posteriores del SDLC, abarcando también los aspectos culturales, además que la seguridad mediante el diseño se ha vuelto más relevante en la digitalización continua de la sociedad, sobre los productos y servicios basados en software y por lo tanto en el ciclo de desarrollo y mantenimiento. Los autores llegaron a la conclusión que, debido a las grandes diferencias entre Agile Scrum y DevOps aumentan la probabilidad de errores, problemas de seguridad y falta de comunicación sustancialmente. DevOps se basa en diferentes principios culturales y otras soluciones técnicas tales como la automatización. Si estos dos aspectos no se abordan durante el desarrollo

pasando luego a la producción podría llegar a ser engorroso. Dada la importancia y el impacto pesar de la seguridad en el software de hoy en día, los autores recomiendan que la seguridad no sea vista como una característica, sino como parte integral del enfoque de desarrollo de software. Y que no sólo los controles duros (contenido y proceso) deben tenerse en cuenta. También controles suaves (relaciones y cultura) deben estar en el foco de una forma equilibrada

De acuerdo a la investigación “Web application development model with security concern in the entire life-cycle” (Macconnel, 1996), desarrollada en el Department of Computer Science and Engineering Yanbu University College, Arabia Saudita. Abordaron el problema de las consideraciones de seguridad de vanguardia en los modelos de desarrollo de aplicaciones web con la necesidad de centrarse en los problemas de seguridad durante todo el SDLC. Los autores tomaron como hicieron una comparación de dos prototipos para realizar su estudio, el primero representando los tipos de consideraciones de seguridad inculcados a través del SDLC web para este estudio. Y el segundo representando el tipo de sistemas en uso en el comercio electrónico hoy en día.

Los autores determinaron que un importante nivel de seguridad es atribuido al efecto de inculcar consideraciones de seguridad durante el Desarrollo del SDLC y que los sitios web mejoran la seguridad si las preocupaciones de seguridad son inculcadas en cada etapa del SDLC web. Por otra parte, los autores señalan que las aplicaciones web de seguridad existentes siguen siendo vulnerables a algún tipo de ataques debido a la falta de consideración de seguridad en todas las etapas del SDLC. Y esto podría atribuirse a la enorme financiación de la tecnología y funcionalidad en lugar de los problemas de seguridad en el desarrollo de aplicaciones web.

En la investigación “HXD: Hybrid XSS detection by using a headless browser” (Choi, Hong, Cho, & Kim, 2018), llevada a cabo en la Universidad de Sejong, se estudiaron las debilidades de las aplicaciones web más recurrentes que se producen cuando una app web realiza una comprobación de entrada deficiente o salida de depuración, demostrando que los detractores pueden usar XSS para proporcionar un script malicioso llevando

la incautación de sesiones, el hurto de credenciales y el aumento de privilegios. Los investigadores plantearon un método de detección de XSS híbrido (HXD), un ángulo de interrupción XSS basándose en una caja negra empleandose tanto para el análisis de cadenas estáticos como la presentación mecánica del navegador. El sistema y enfoques presentado por los autores ejecutaron pruebas HXD en las aplicaciones web más importantes del portal de internet de Corea “Never”. Como conclusión consideraron que la evaluación HXD tiene bajo porcentaje de dar falsos negativos.

Según “XSS Vulnerability Detection Using Optimized Attack Vector Repertory” (Guo, Jin, & Zhang, 2015)), realizada en Instituto de Tecnología de la Computación y en la Academia China de Ciencias, se trató el tema de fragilidades de XSS en las aplicaciones web, en las que evidencian que la gran mayoría de aplicaciones web experimentan vulnerabilidades de tipo XSS. Los autores plantean una forma de detección de vulnerabilidad XSS usando una lista ideal de vectores de ataque. Este sistema genera una lista de vectores de ataque espontáneamente, perfecciona la lista de vectores de ataque usando un sistema de optimización, detectando las vulnerabilidades de XSS en aplicaciones web activamente. Para mejorar la lista de vectores de ataque, se fabrica un modelo de perfeccionamiento con un algoritmo de aprendizaje automático, disminuyendo la lista de vectores de ataque y optimizando la capacidad de la detección de vulnerabilidades XSS. Llegando a la conclusión de que el método da buenos frutos en la optimización del repertorio de vectores de ataque.

En el estudio “Reverse analysis method of static XSS defect detection technique based on database query language” (Baojiang, Baolian, & Tingting, 2014), llevado a cabo en la Universidad de Beijing de Correos y Telecomunicaciones. Estudiaron el tema de seguridad habitual en aplicaciones web, la vulnerabilidad de XS, demostrando la gravedad de la pérdida de información en los usuarios. Los investigadores presentaron un sistema de evaluación estacionario de la detección de carencias XSS de la aplicación web java mediante el análisis contrapuesto de la serie de datos. El método se probó a través de intentos en proyectos web Java creados

aparentemente con errores XSS y unos cuantos proyectos web Java de código abiertos. Llegando a la conclusión de que el método no solo perfeccionó la eficiencia de detección, sino que además mejoró la exactitud de detección para la falla XSS.

De acuerdo a la investigación “Automatic web security unit testing” (Mohammadi, Chu, Lipford, & Murphy, 2016), abordado en el taller Internacional IEEE / ACM en Automatización de Pruebas de Software. Observaron la incorporación de las evaluaciones de seguridad en el flujo de trabajo de los desarrolladores de software, no solo ahorrando recursos para realizar pruebas de seguridad apartademente, sino que además disminuye el costo de la respuesta de la seguridad, demostrando que la mayoría de vulnerabilidades se descubren al inicio del ciclo de desarrollo. Los investigadores plantean un enfoque de evaluaciones automáticas para la detección de XSS, originadas por la errónea codificación de datos no fidedignos. Logrando concluir que el enfoque afianza la comprobación de entrada delante de una inyección de línea de comando.

En la investigación “Detecting cross-site scripting vulnerability using concolic testing” (Ruse & Basu, 2013), los investigadores discutieron sobre el problema de ataques de XSS en la web, realizandose una secuencia de comandos insidiosa (desde una fuente acopiada), demostrando que se usa para el hurto de información, obteniendo ingreso desacreditado a los medios del usuario y del sistema. Los autores plantean una técnica de dos etapas para detectar las vulnerabilidades de XSS y anticipar los ataques de XSS. En la primera etapa, convierten la aplicación web a un idioma en el cual las herramientas de prueba concólica desarrolladas recientemente estén aptas. En la segunda etapa, en función de las necesidades de entrada / salidas establecidas en la primera etapa, se instrumentan automáticamente el código correcto de la aplicación a través de la integración de monitores. El sistema planteado por los autores, determina las vulnerabilidades XSS que se producen a causa de la copia de entradas a salidas y la edificacion de entradas en serie malintencionadas partiendo de la unión de entradas particularmente beneficiosas. Llegando a la conclusión de que el alcance del

prototipo de inserción del Framework es propicio, usando aplicaciones web JSO sustanciales.

1.3. Abordaje teórico

1.3.1. Software

Según el estudio (M., 2010, pág. 10), define el software como varios programas individuales que manejan datos ya sean comerciales o técnicos, ayudando asi a que las operaciones del negocio se agilicen y mejoren con el fin de resolver las necesidades del negocio.

1.3.2. Ingeniería de Software

La ingeniería de software comprende procesos complejos que van desde la especificación del proyecto, hasta la puesta en marcha y mantenimiento que requiere después de su uso, considerando el uso de herramientas y métodos para dar soluciones automáticas a problemas reales. (M., 2010) (Loic, 2005)

1.3.3. Proceso de desarrollo de software

De acuerdo con Ivar et al. (Díaz, Cerro, & Proenza, 2014)El proceso se basa en controlar y gestionar distintos recursos que se usan para alcanzar los objetivos planteados, el cuál guiará a los participantes durante todo el desarrollo del software con el propósito de crear un producto eficaz y eficiente que atiendan lo que el cliente desee.

Para el desarrollo de software se utilizará una estructura, donde existen 4 actividades importantes del proceso, que son la especificación y definición de las funcionalidades y restricciones del software; el desarrollo del diseño del software acorde con las especificaciones; comparación y validación del sistema y, la evolución y modificación del software de acuerdo a las modificaciones requeridas por el usuario y entorno exterior. (Díaz, Cerro, & Proenza, 2014, pág. 12)

1.3.4. SDLC

De acuerdo al estudio (Kaisler, 2005, pág. 20), el SDLC es un modelo que establece el inicio u final del proyecto, estableciendo el orden del proceso, y

su definición es primordial para establecer el plan del proyecto para consecución del objetivo.

De acuerdo a (Kaisler, 2005, pág. 25), el SDLC también implica definir la relación entre cada tarea que forme parte del plan de desarrollo del modelo, por lo que es importante tener en cuenta cinco tareas principales, el modelado, la planeación, la comunicación, el despliegue y la construcción.

1.3.5. Modelos de procesos para el desarrollo de software

Son las fases o actividades que se deben realizar, así como en qué orden se desarrollan. Como se afirma en (Shuaibu & Ibrahim, 2018, pág. 12) seleccionar erróneamente un modelo de SDLC provocaría retrasos en el desarrollo de software. Lo mismo ocurriría si no se trabaja con un modelo de SDLC para desarrollar un proyecto de software, ya que no habría una guía y se podría trabajar doble o innecesariamente. Existen varios modelos de SDLC, los cuales son mostrados a continuación.

1.3.5.1. Modelo cascada

En este modelo de SDLC el proyecto de software se realiza una cadena de fases, empezando con la noción original del sistema hasta las últimas evaluaciones del software. En este modelo de desarrollo es necesario realizar revisiones al finalizar cada etapa, para asi poder identificar cualquier deficiencia que impida continuar con la siguiente etapa.

La modelo cascada es muy útil para identificar problemas en las etapas iniciales del desarrollo, “funciona correctamente cuando los requerimientos de calidad dominan sobre los requerimientos de coste y de planificación”. Ver (Loic, 2005, pág. 16).

De acuerdo al estudio de (Shuaibu & Ibrahim, 2018, pág. 19), el modelo tiene las siguientes desventajas: se espera que los requerimientos no varíen durante el proceso de desarrollo, lo que es muy poco probable, los límites entre cada una de sus fases son muy rígidos, esto conlleva a que la revisión de los hitos y entregables entre cada fase sea complicado, lo que detiene todo el desarrollo del proyecto, las fallas y errores de cada una de las fases se detectan recién en la etapa de verificación, es decir cuando todo el código

ya se ha desarrollado y provocaría que reparar estos problemas requiera mucho tiempo, esfuerzo y dinero.

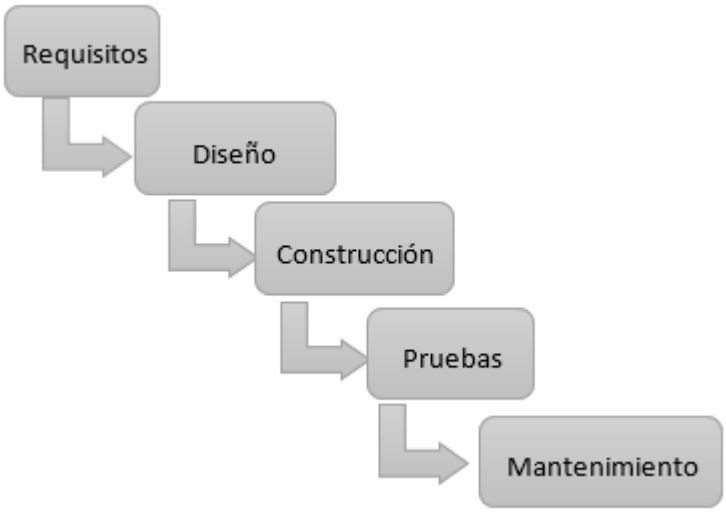


Figura 2 Representación gráfica del SDLC en cascada. Ingeniería del software un enfoque desde la guía (Sánchez, Sicilia, & Rodríguez, 2012)

1.3.5.2. Modelo proceso en "V"

En el apartado (Shuaibu & Ibrahim, 2018, pág. 22)señala que este modelo es una versión mejorada del modelo cascada, en el cual se prioriza la validación y verificación del software.

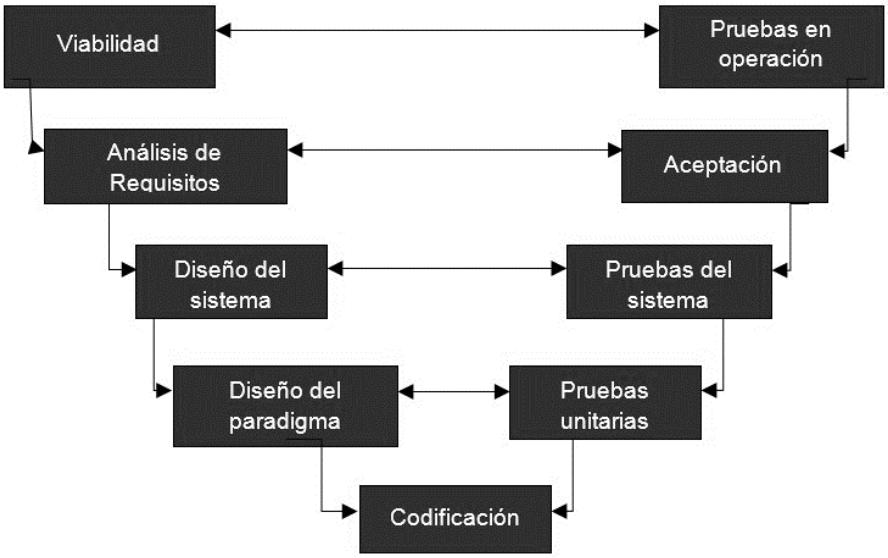


Figura 3 Representación del SDLC en "V". (Sánchez, Sicilia, & Rodríguez, 2012, pág. 41)

De acuerdo a otro autor, se explica quel a relación entre los procedimientos de aseguramiento de la calidad y los relacionados con la comunicación, el modelado y la construcción. A medida que el equipo de software desciende por el lado izquierdo de la V, los requisitos previos para el problema mejoran en representaciones técnicas más detalladas del problema y su solución. Iniciando desde la codificación, en el lado derecho están las pruebas unitarias, las cuales están relacionadas con el diseño de bajo nivel, y en otros casos son las pruebas del código. La siguiente etapa del lado derecho señala las pruebas del sistema, en esta fase se realizan las pruebas en la estructura de alto nivel. Luego continúan las pruebas de aceptación, donde se evalúan los requerimientos definidos correspondan a las necesidades del cliente (Loic, 2005, pág. 24).

1.3.5.3. Modelo espiral

En la investigación (Sánchez, Sicilia, & Rodríguez, 2012, pág. 10), se grafica el proceso de desarrollo de sistemas como un espiral, además explica que el modelo fue publicado en 1988 por Boehm. Cada ciclo del espiral es una etapa del desarrollo de software, en el cual el ciclo interior se refiere a lo viable que pueda ser el desarrollo del software, el siguiente a la aclaración de requerimientos, el que sigue al diseño, etc., El gráfico que representa al modelo en espiral está partido en cuatro cuadrantes, los cuales definen actividades diferentes, los cuales son la definición de objetivos, en el cual se definen objetivos específicos y se identifican restricciones, se lleva a cabo la identificación de riesgos y planes para contrarrestarlos. La reducción de riesgos es donde se realiza un análisis a detalle de los riesgos presente en el proyecto, de tal forma que se puedan definir estrategias para reducirlos. En la ingeniería, es decir el desarrollo y validación, es necesario construir prototipos evolutivos para el desarrollo. El planeamiento, es donde se evalúa el estado del software, para así decidir seguir con el siguiente ciclo de la espiral. En caso se decida seguir con el proyecto, se deben desarrollar planes para la siguiente etapa del desarrollo.

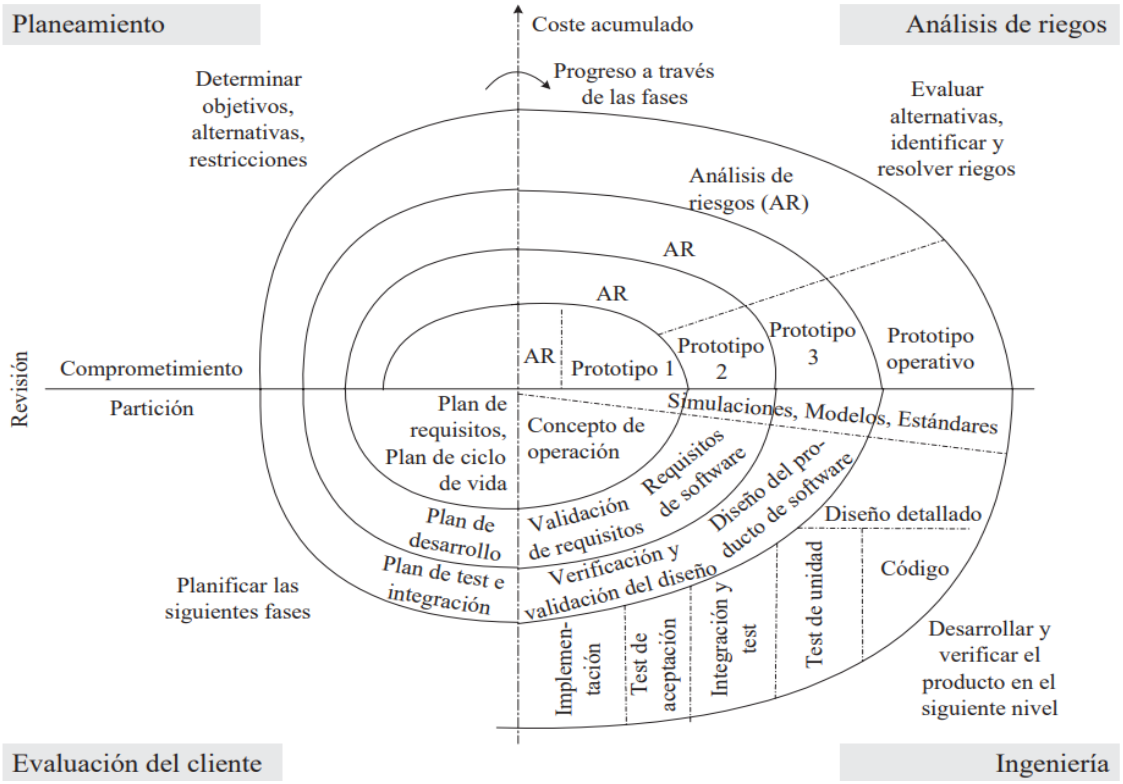


Figura 4 Representación gráfica del SDLC en espiral. (Pressman, 2010)

Así mismo, (Sánchez, Sicilia, & Rodríguez, 2012, pág. 12), se menciona que este modelo de procesos contiene las mejores características en cuanto al uso de prototipos. Siendo uno de sus puntos fuertes el hecho de que el software se desarrolle de manera progresiva, ayuda a la cooperación con los clientes, lo cual lleva a la aprobación progresiva del sistema.

1.3.5.4. Modelo de proceso incremental

El apartado (Sánchez, Sicilia, & Rodríguez, 2012, pág. 15), menciona que es mucho mejor utilizar procesos incrementales cuando se necesita entregar al usuario funcionalidades del sistema rápidamente e ir aumentando estas funcionalidades en cada entrega de nuevas versiones del sistema.

Si se decide usar este modelo es necesario tener presente qué proceso vital se va a desarrollar en el sistema, ya que asi el usuario puede probar las primeras versiones e ir solicitando cambios o mejoras que serán presentadas posteriormente en nuevas versiones del producto de manera incremental, también será necesario realizar un plan que defina las acciones a tomar para desarrollar las siguientes mejoras, teniendo en cuenta subsanar errores

encontrados en la versión anterior, asi como también añadir nuevas funciones basadas en los requerimientos solicitados por el cliente. Cada mejora en el sistema base se debe presentar como un producto final operable. Todo este proceso se debe repetir hasta finalizar el sistema.

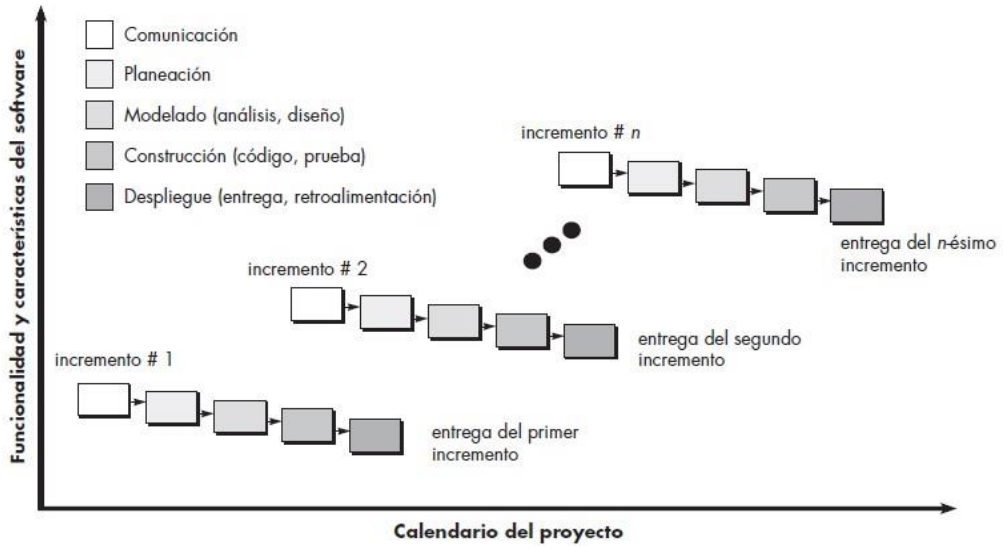


Figura 5 Representación gráfica del SDLC incremental. (Pressman, 2010)

1.3.6. Metodologías.

La metodología de desarrollo de software hace referencia a un framework que se usa para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información, y a lo largo del tiempo se ha diferenciado de acuerdo a sus distintas fortalezas y debilidades y por el objetivo principal que persiguen, ser ágiles y robustas. Debido a esto plantean un proceso disciplinado, con el fin de hacerlo más predecible y eficiente, para así aumentar la calidad del software que se produce en cada una de las fases del desarrollo.

1.3.6.1. Metodologías para el desarrollo ágil del software.

En la actualidad uno de los requerimientos más críticos es el desarrollo y entrega rápido de los sistemas. Las organizaciones prefieren perder calidad en el sistema para asi obtener un software en el menor tiempo posible.

La metodología ágil busca satisfacer las necesidades del cliente con una entrega rápida del software base, que irá mejorando de manera incremental con cada entregable que se presente, se trabaja con equipos pequeños de desarrolladores que utilizan métodos informales y simplifican en gran medida

el desarrollo del sistema. Estos procesos para el desarrollo rápido del software han sido diseñados para generar sistemas útiles de la manera más rápida posible. Usualmente se usan procesos que combinan el diseño con el desarrollo y pruebas.

Es necesario tener en cuenta que estos métodos ágiles no deben ser utilizados para desarrollar sistemas críticos donde es completamente necesario generar primero un análisis detallado de los requerimientos del sistema para poder identificar qué medidas de seguridad deben ser aplicadas. El uso metodologías ágiles se ha visto incrementado de manera poco usual en la industria, ya que según el Cutter Consortium, el 50% de las empresas desarrolladoras afirman que al menos la mitad de sus proyectos se desarrollan con metodologías ágiles. Algunas de estas metodologías más usadas actualmente son las siguientes.

1.3.6.1.1. Metodología XP programación extrema

La metodología extrema (XP) es probablemente el método ágil más utilizado y conocido. Algunos consideran que fue una de las mejores innovaciones mientras que otros lo rechazan (Ramos, 2015). En la metodología XP los requerimientos planteados son llamados historias de usuario, y son definidos como serie de tareas a realizar. Los desarrolladores trabajan en pares y crean pruebas para cada tarea asignada antes de comenzar a programar. Estas pruebas se deben poder ejecutar sin problemas cuando se termine la fase de programación y se una con el sistema base para asi poder presentar la nueva versión.

Como en toda metodología ágil se utiliza el desarrollo incremental el cual se trabaja de manera que cada cierto tiempo se presentan pequeños entregables del sistema enfocados en los requerimientos del usuario, es decir, en las historias de usuario, los cuales son la base para planificar avances.

1.3.6.1.2. Metodología SCRUM

SCRUM es una metodología ágil que se utiliza para desarrollar software, la cual tiene como objetivo aumentar considerablemente el retorno de la inversión. Principalmente se baja en desarrollar la funcionalidad de mayor

importancia para el usuario, adoptando principios como innovación y mejora continua.

SCRUM le permite al cliente cambiar los objetivos del negocio en cualquier momento, ya que puede incluir cambios en la funcionalidad o cambiar prioridades al iniciar cada iteración. De esta forma se prioriza la innovación y promueve el compromiso de trabajo en equipo para el proyecto. Esta facilidad de adoptar cambios, aumenta la capacidad de reacción ante cambios inesperados realizados por el cliente adaptándose a cualquier tipo de cambio que exista en el proyecto sin importar la complejidad del mismo. Gracias a esto los clientes pueden dar uso a los entregables con las funcionalidades deseadas en un menor tiempo, garantizando asi la calidad en el software, ya que se podrá ir mejorando en cada iteración.

1.3.7. Seguridad en ingeniería de software

Según el estudio realizado por los autores Díaz, Cerro y Proenza, *(Díaz, Cerro, & Proenza, 2014, pág. 13)*, la seguridad de software es aplicar la seguridad de información al desarrollo de aplicaciones. La seguridad de información es la protección de los sistemas contra el acceso no autorizado o la modificación de datos cuando están en la etapa de procesamiento, transito o almacenamiento.

También la seguridad de información debe proteger el software contra la negación de servicios, además de medidas necesarias para detectar y documentar tales ataques, para asi poder tomar acciones.

Las interrogantes más comunes con respecto a la seguridad de software están relacionadas con el SDLC. La respuesta más común es que la seguridad debe ser considerada durante la fase de diseño y desarrollo, además la seguridad debe mantenerse presente durante el uso y mantenimiento del software para asi asegurar su integridad.

La gran cantidad de seguridad aplicada en los Sistemas de Redes actuales, pueden hacer creer a los desarrolladores de software que ya no necesitan aplicar seguridad a los sistemas. Sin embargo, esto es completamente errado, ya que un sistema sin seguridad está expuesto a muchos tipos de ataques que la Seguridad de Redes no cubre. Aquí se

presentan dos problemas, los softwares que en teoría son seguros, pueden ser inseguros en la etapa de producción, y los softwares que van creciendo y se hacen más complejos, aumenta la posibilidad de nuevos fallos lo que aumenta la oportunidad de atacarlos.

1.3.8. Seguridad de software

La definición de seguridad en el software según un estudio realizado (Schwaber & Sutherland, 2011, pág. 14), afirma que es un punto muy importante de la Ingeniería de Software, ya que el uso de internet se ha incrementado muy rápidamente, como es el comercio electrónico. Esto quiere decir que la probabilidad de ataques a aumentado considerablemente, como así también han aumentado las consecuencias negativas que estos ataques pueden ocasionar.

Actualmente todo sistema debería incorporar seguridad para poder tener una forma de defenderse de la mayoría de ataques. Los desarrolladores deben concentrarse, además de en los requerimientos del cliente o usuario, en las posibles brechas de seguridad en el sistema.

Esto ha logrado generar unos cambios en el diseño y desarrollo del sistema, para poder agregar la seguridad en los requerimientos, además de los definidos por el cliente.

Como es de esperarse, todo gran cambio requiere de tiempo, ya que no se puede lograr de un día para otro, pero estos son los primeros pasos para que los desarrolladores de software apliquen la seguridad de software en sus desarrollos. Uno de los posibles problemas es el hecho de que la Ingeniería de Software no ha dado una solución eficaz y aplicable a la falta de seguridad en el software, muchas de las opciones actuales tienen fallas o debilidades por el mismo hecho de que es un proceso que se encuentra recién desarrollándose.

1.3.9. Tipos de software de seguridad

De acuerdo al estudio realizado (Schwaber & Sutherland, 2011, pág. 16), se menciona que el tipo de softwares más conocido en cuanto a la seguridad, son los antivirus, programas que ofrecen protección y eliminación de virus informáticos. La mayoría de software de antivirus tienen una base de datos

de firmas de virus que se actualiza constantemente para asi poder estar al día con cada virus nuevo que aparezca.

Actualmente la seguridad es necesaria en todo sistema informático. Pero uno de los problemas para implementar la seguridad es el hecho que la Ingeniería de Software todavía no bridan un mecanismo eficaz para implementar la seguridad, ya que los lenguajes de programación suelen tener código muy complejo y confuso cuando de seguridad se está hablando.

1.3.10. Seguridad en el ciclo de desarrollo del software

Como suele suceder, la mayoría de empresas contrata sus propios desarrolladores para realizar su propio aplicativo, un software dedicado a sus necesidades del negocio. Como en todo software, las aplicaciones pueden tener problemas de seguridad, pero a diferencia de un software comercial, estos sistemas realizados por la misma empresa, no disponen de actualizaciones periódicas ni de parches proporcionados por el fabricante. Entonces solucionar estos problemas es responsabilidad de la misma organización que desarrolló su sistema. (Kaisler, 2005, pág. 17)

Una de las malas prácticas muy común en las empresas es poner en producción un sistema que no ha sido testeado por el sector de Seguridad de la Información.

En otras ocasiones el sector de Seguridad de la Información se quiere aplicar al finalizar el desarrollo lo cual dificulta el tomar acciones para integrar seguridad en la aplicación ya desarrollada.

Generalmente, en el mejor de los casos, se propone un testeo de seguridad cuando la aplicación ya está completamente terminada. Aquí es donde recién se descubren errores de seguridad que requieren el rediseño de una o varias partes del sistema, lo que se resume en un aumento en tiempo, esfuerzo y dinero.

1.3.11. Seguridad en el análisis de requerimientos

En este punto se deben definir los requerimientos funcionales que tendrán impacto en la seguridad de la aplicación. Estos pueden ser el tipo de información que se procesará, por ejemplo, datos personales, financieros, pagos electrónicos, información pública o confidencial, contraseñas, etc. El cumplimiento con normativas como por ejemplo SOX, “A” 4609, PCI, etc.

1.3.12. Seguridad en el diseño

Para poder comenzar a programar, antes de eso se debe tener en cuenta muchos aspectos de seguridad al momento de realizar el diseño del software.

Según una investigación. Ver (Shuaibu & Ibrahim, 2018, pág. 7). Algunos aspectos de seguridad son: Diseño de autorización, aquí se deben definir permisos, privilegios y roles del sistema. Diseño de autenticación, en este punto se diseña la forma en que el usuario se va a autenticar en el sistema, donde se incluyen mecanismos de autenticación como las contraseñas, certificados, tokens, etc. y los mecanismos de protección que tendrá el sistema para evitar cualquier tipo de ataque como los de fuerza bruta o diccionario, un ejemplo sería el uso del captcha. Diseño de mecanismos de protección de datos, aquí se analiza y define como se va a proteger los datos sensibles almacenados en la aplicación, como por ejemplo el uso de hashes para encriptar la información.

1.3.13. Seguridad en la codificación

Cuando el diseño del software está completado, los desarrolladores deben comenzar a codificar los componentes de la aplicación. Aquí es donde suelen agregarse por error o desconocimiento distintos tipos de vulnerabilidades en el software. Estas vulnerabilidades se dividen en dos grupos, las clásicas y las funcionales. Las clásicas son conocidas y están categorizadas.

Varios ejemplos de estas vulnerabilidades son las mostradas en el “OWASP

Top 10” (Cross Site Scripting, Inyección, Entidades externas XML, etc) asi como también otras vulnerabilidades que no están directamente ligadas con el desarrollo web, como es la denegación de servicio, desbordamiento de buffer, etc. Los Frameworks actuales de desarrollo son una muy buena ayuda con estos problemas comunes, ya que se encuentran entre el código y el programador, previniendo asi la mayoría de vulnerabilidades ya conocidas.

Las vulnerabilidades funcionales son las que están relacionadas específicamente con la funcionalidad del negocio, por lo que no están categorizadas.

Unos ejemplos de las vulnerabilidades funcionales pueden ser: un sistema de venta para entradas de cine que no limita a los usuarios la cantidad de reservas, una banca virtual que permite realizar depósitos, retiros y transferencias con montos negativos, etc.

En esta etapa de codificación, uno de los detalles más importantes es verificar cada valor de entrada y salida, teniendo en cuenta que es posible que se ingresen datos maliciosamente para intentar atacar o hacer fallar el sistema.

1.3.14. Confiabilidad del software

Al pasar de los años, los sistemas suelen llegar a ser cada vez más robustos y complejos, lo que aumenta la probabilidad de que ocurran errores en el sistema.

Es muy difícil demostrar que un sistema es seguro, ya que un software seguro debe poder funcionar sin problemas y sin comprometer datos aun estando bajo un ataque. (Sánchez, Sicilia, & Rodríguez, 2012, pág. 5)

Es conocido que casi todos los sistemas tienen errores, pero la mayoría de estos nunca son encontrados bajo circunstancias normales, esto es lo que un atacante busca, encuentra la debilidad y ataca al sistema.

Las empresas que desarrollan software deben utilizar en mayor o menor parte las prácticas efectivas que permiten mejorar la calidad del software. Una de estas prácticas efectivas es la Ingeniería de la Confiabilidad de Software, la cual es una práctica cuantitativa que puede ser integrada en empresas bajo cualquier modelo de desarrollo sin importar el tamaño de la misma.

Las empresas desarrolladoras de software destinan muchos recursos en incrementar la calidad de sus productos. Una parte de estos recursos se usa para adoptar mejores prácticas, pero la dificultad que existe para adoptar estas prácticas no es sólo el tiempo elevado que conlleva aplicarlas o su costo, sino el medir su nivel de impacto real en la calidad del producto, así como indicar la reposición de tal inversión.

La calidad de un producto o servicio es percibida por los usuarios o clientes de una empresa. En el caso de las empresas desarrolladoras de software, la calidad de su servicio se percibe en cuantas fallas detecta el cliente a la hora del uso del mismo.

La confiabilidad mide el nivel en que un producto funciona sin problemas bajo condiciones específicas por un periodo de tiempo determinado. Esta es un atributo cuantitativo que ha sido analizado, estudiado y usado en muchas industrias para evaluar la calidad de un producto o servicio.

Una falla en el sistema es percibida por el cliente como algo que no funciona correctamente y genera desconfianza, y símbolo de una baja calidad. En cambio, un defecto es el problema existente en el software, es decir, lo que genera la falla.

Un software es confiable cuando realiza lo que el usuario desea en el momento que lo necesite, deja de ser confiable si no cumple con lo anterior indicado. En resumen, un software no es confiable cuando falla. Estas fallas se deben a defectos en el software. Entonces si se corrigen estos defectos sin agregar nuevos, se mejora la confiabilidad del software, lo que mejora su calidad. (Sánchez, Sicilia, & Rodríguez, 2012, pág. 8)

Antiguamente, una forma de aumentar la confiabilidad de un sistema era ejecutarlo y probarlo por un tiempo prolongado de tiempo antes de liberarlo para asi poder encontrar cualquier falla. Esto actualmente no es efectivo, ya que no se debe probar la confiabilidad en el producto, sino crearla, es decir, la confiabilidad debe ser diseñada en el software.

1.3.15. Ingeniería de Seguridad

De acuerdo a un estudio de Ingeniería. Ver (Sánchez, Sicilia, & Rodríguez, 2012, pág. 10), la Ingeniería de la seguridad usa varias ciencias para diseñar características de seguridad, sistemas de seguridad y controles en general. La motivación principal de esta ingeniería es el evitar cualquier tipo de ataque.

Esta ingeniería posee un campo bastante amplio, comenzando por los equipos, con el diseño de cerraduras, configuración de cámaras de seguridad, sensores especiales; también en procesos como es el control de acceso o en la informática con el control de contraseñas y la criptografía.

En la actualidad la seguridad está relacionada con la interacción con otros sistemas. Esto quiere decir que la seguridad de un sistema en particular se encuentra relacionada directamente con la seguridad de los otros sistemas relacionados.

En cuanto a la seguridad por contraseñas y mecanismos criptográficos, muchos sistemas pueden poseer los mejores mecanismos, los más complejos y seguros, pero si su debilidad se encuentra en otra parte del programa, no va a ser necesario atacar el sistema criptográfico.

Un ejemplo de esto serían los sistemas vulnerables debido a problemas en el código, los cuales pueden ser atacados con "buffer overflow" al no tener en cuenta aspectos importantes como la excepción en el sistema, lo que llevaría a que un atacante hábil obtenga permisos de administrador. Esto llevaría a que, si una persona logra obtener este nivel de permisos, se le facilitará mucho conseguir las claves de acceso, lo que dejaría completamente inútil la fortaleza del sistema criptográfico utilizado para proteger el sistema. Muchos atacantes suelen aprovecharse de fallas en el diseño de los sistemas, por ejemplo, si una aplicación ejecuta comandos sin evaluar ciertos niveles de autenticación, el atacante puede conseguir información importante sin tener que intentar romper las contraseñas que protegen la información.

En la actualidad, gracias a la comunicación móvil, muchos sistemas utilizan las conexiones inalámbricas, lo cual facilita que espíen la señal. Los ataques de ingeniería social suelen ser muy efectivos en el momento de atacar sistemas que aparentan ser seguros.

La falta de conocimiento de algunos desarrolladores de sistemas sobre los mecanismos de seguridad implícitos ocasiona que los sistemas sean más vulnerables.

1.4. Formulación del Problema

¿Cómo desarrollar software con características de seguridad para vulnerabilidades más recurrentes?

1.5. Justificación e importancia del estudio

La investigación sobre un nuevo modelo de procesos para el desarrollo de software para implementar seguridad en todas las etapas del SDLC, permite que las empresas desarrolladoras reduzcan significativamente sus gastos en tiempo y dinero revisando fallas de seguridad en el software en la etapa final, logrando así mayor productividad y reducir costes.

Este nuevo modelo sirve de base para el desarrollo de software seguro y puede ser modificado de acuerdo a las necesidades de la empresa o para crear nuevos modelos de procesos con atributos de calidad de software añadidos.

1.6. Objetivos

1.6.1. Objetivo general

Desarrollar un modelo de procesos para el desarrollo de software con características de seguridad para vulnerabilidades más recurrentes.

1.6.2. Objetivos específicos

a. Identificar las vulnerabilidades de seguridad más recurrentes y seleccionar métodos a partir de modelos existentes y estrategias para la seguridad en el desarrollo de software.
b. Integrar los métodos, estrategias y buenas prácticas en un modelo de procesos.
c. Validar el modelo propuesto.

1.7. Limitaciones

1.7.1. Limitaciones culturales en las empresas

Las empresas pequeñas dedicadas al desarrollo de software no suelen darle la importancia debida a la seguridad en sus procesos de desarrollo de software

1.7.2. Acceso a pruebas y comunicación

Por la coyuntura actual, las empresas no dieron la oportunidad de aplicar el modelo de procesos que se ha desarrollado en esta investigación. Además de la dificultad de comunicación a profesionales expertos en seguridad de software debido a la pandemia.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación

El tipo de investigación de este proyecto es cualitativo y el diseño de investigación es no experimental.

2.2. Escenario de estudio

El escenario de estudios son las empresas desarrolladoras de software en Chiclayo.

2.3. Caracterización de sujetos

Los sujetos fueron los expertos que basados en su opinión y experiencia se pudo saber si el modelo de procesos desarrollado fue el correcto para las empresas peruanas desarrolladoras de software

2.4. Técnicas e instrumentos de recolección de datos

2.4.1. Técnicas

Someteremos nuestro modelo a una prueba de modo juicio de expertos mediante el método Delphi mediante encuestas enviadas a tres individuos aleatorios expertos en el tema.

2.4.2. Instrumentos

Juicio de Expertos (Método Delphi).

2.5. Procedimiento para la recolección de datos

El presente proyecto de investigación tiene un enfoque cualitativo, en el cual se aplicó el método Delphi para su validación, se procedió a seleccionar expertos en seguridad informática, luego se preparó un instrumento el cual fue un cuestionario, y se eligió la vía de consulta, es este caso por medio del correo electrónico.

2.6. Procedimiento de análisis de datos

Luego de haber recolectado la información se revisaron las respuestas regostradas y se enviaron de nuevo para una retroalimentación y consolidación de datos, finalmente se generó un reporte de resultados.

2.7. Criterios éticos

a. **Veracidad:** El entorno en que se llevaron a cabo las evaluaciones sea objetivo y no subjetivo, donde los resultados fueron acordes a lo implantado.
b. **Confidencialidad:** El estudio vigilió la ética profesional al momento de la ejecución de las evaluaciones, proveyó privacidad a la información que controlaban los servidores.
c. **Derechos de Autor:** Todo documento utilizado para esta investigación estará citados junto con sus autores respectivamente.

2.8. Criterios de Rigor Científico

a. **Fiabilidad:** Las técnicas utilizadas para validar la eficacia del modelo, nos permitirá determinar qué tan viable es aplicarlo en una empresa.
b. **Consistencia:** Toda la información obtenida para esta investigación es de carácter científico y formal. El estudio realizado a los datos se ha realizado con total profesionalidad para mantener asi la veracidad de los datos y esto resulte en información útil y consistente.
c. **Originalidad:** Se citará las fuentes bibliográficas, con el objetivo de demostrar la inexistencia del plagio.

III. REPORTE DE RESULTADOS

3.1. Análisis y Discusión de los resultados

Realizando un consenso de respuestas de los evaluadores del modelo de procesos aplicando el método Delphi con aporte del análisis del autor de esta investigación, se puede afirmar lo siguiente:

El modelo propuesto es viable para las pequeñas empresas desarrolladoras porque contiene las fases básicas y esenciales que todo proceso de desarrollo de software debería poseer, además de ser de fácil comprensión para desarrolladores promedio y avanzados.

El modelo propuesto es de bajo coste de implementación ya que se utilizan herramientas y procedimientos de libre acceso, los cuales facilitan su implementación, aunque en un inicio es probable que aumente el tiempo del desarrollo de software, su beneficio será notable en la etapa de producción

al haber reducido vulnerabilidades de seguridad, lo cual se traduce a evitar rehacer o agregar código al software en post producción.

Las herramientas automatizadas para realizar las pruebas estáticas y dinámicas reducen el tiempo del desarrollo, pero no se pueden estimar valores numéricos sobre el costo-beneficio ya que no se ha realizado un estudio a ese nivel.

Los procesos definidos en el modelo propuesto cumplen con asegurar la seguridad básica y necesaria en el desarrollo de software aplicado a pequeñas empresas desarrolladoras. Pero se debe tener en cuenta el alcance del software a desarrollar porque en casos especiales, existe software propenso a ataques, los cuales requerirían agregar más procesos y fases al modelo propuesto para apuntar directo a esos requerimientos especiales.

El modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software ya que contiene lo esencial del SDLC-S ayudando así a los desarrolladores a no pasar por alto la seguridad en ninguna de las etapas del desarrollo de software.

El modelo propuesto se puede adoptar por la mayoría de metodologías de desarrollo de software tradicionales, pero no para metodologías más actuales, como son las ágiles.

Los productos resultantes en cada etapa del modelo propuesto son de ayuda para los programadores a cargo para asi mantener un registro de sus acciones realizadas en cuanto a seguridad, y una guía sobre qué acciones tomar a continuación en el transcurso del desarrollo del software.

Este estudio tuvo por objetivo principal impulsar un modelo de procesos para el desarrollo de software con cualidades de seguridad para vulnerabilidades más recurrentes, para ello se identificaron cuáles eran las vulnerabilidades de seguridad más comunes, usando como referencia principal la investigación realizada por OWASP, la cual hace mención de los diez riesgos más críticos en las aplicaciones web.

Asi mismo se pasó a seleccionar métodos y estrategias para la seguridad en el desarrollo de software y evitar las vulnerabilidades anteriormente

identificadas. Con ayuda del artículo publicado por OWASP se pudo encontrar soluciones específicas para cada vulnerabilidad, las cuales se seleccionaron para ser posteriormente organizadas.

Seguido de ellos se procedió a integrar estos métodos, estrategias y buenas prácticas definidas anteriormente para crear el modelo de procesos orientado a seguridad del software, organizándola en etapas.

Finalmente se validó el modelo propuesto con el juicio de expertos.

Todo el progreso de los objetivos mencionados, se describen con más detalle en los puntos siguientes.

3.1.1. Identificar las vulnerabilidades de seguridad más recurrentes y seleccionar métodos a partir de modelos existentes y estrategias para la seguridad en el desarrollo de software.

Las vulnerabilidades de seguridad más recurrentes fueron obtenidas gracias a las investigaciones de OWASP en su documento OWASP TOP 10 – 2017 Los diez riesgos más críticos en Aplicaciones Web, el cual selecciona y ordena por mayor a menor riesgo cada vulnerabilidad, gracias a sus investigaciones previas en fechas anteriores y su importancia en el ámbito de seguridad en el desarrollo de software.

Tabla 2

Lista de riesgos de seguridad más recurrentes en aplicaciones junto con su descripción.

Riesgos de Seguridad en Aplicaciones	**Descripción**
Inyección	La inyección SQL se produce cuando se envían datos no confiables para el intérprete, como parte de un comando o consulta. El atacante puede engañar al intérprete para ejecutar los comandos involuntarios o acceder a los datos sin permiso.

Pérdida de Autenticación	La funcionalidad relacionada con la autenticación y la gestión de sesiones se implementa incorrectamente, lo que permite a los atacantes comprometer a un usuario y contraseña, tokens de sesión o errores de explotación realizados para que el atacante pueda asumir la identidad de otro usuario (temporal o permanente)
Exposición de datos sensibles	Muchas aplicaciones web y API protegen los datos confidenciales incorrectamente, como la información financiera, la información médica o de identificación personal. Los atacantes pueden robar o modificar estos datos causando fraude de tarjetas de crédito, robo de identidad u otros delitos. Los datos confidenciales requieren métodos de protección adicionales, como el almacenamiento y el cifrado de tráfico.
Entidades externas XML (XEE)	La mayoría de procesadores XML antiguos o mal configurados evalúan las referencias a entidades externas en documentos XML. Las entidades externas se pueden usar para divulgar archivos internos con URI o archivos internos en servidores mal configurados, encontrar puertos abiertos, ejecutar comandos bash y realizar ataques DoS.
Pérdida de control de Acceso	Las limitaciones sobre lo que los usuarios no pueden hacer no se aplica de manera correcta. Los atacantes pueden aprovechar estos errores de acceso, de manera deshautorizada, acceder a funciones y / o datos, cuentas de otros usuarios, consulte los archivos confidenciales, modificar los datos, modificar permisos, etc.
Configuración de Seguridad Incorrecta	La configuración de seguridad incorrecta es un problema muy común y se debe a una configuración manual, ad hoc o predeterminado (o directamente de la falta de

	configuración). Por ejemplo: los encabezados HTTP están mal configurados, los mensajes de error con contenido sensible, falta de parches y actualizaciones, marcos de trabajo, dependencias y componentes obsoletos, etc
Secuencia de Comandos en Sitios Cruzados (XSS)	XSS ocurre cuando una aplicación toma datos no confiables y los envía al navegador web sin autenticación y cifrado precisos; O actualiza un sitio web existente con los datos proporcionados por el usuario utilizando la API de ejecución de JavaScript en el navegador. Le permiten realizar comandos en el navegador de la víctima y un atacante puede redirigir una sesión, cambiar el sitio web o redirigir al usuario a un sitio web malicioso.
Deserialización Insegura	Esto ocurre cuando una aplicación recibe objetos serializados maliciosos y estos objetos se pueden manipular o eliminar por el atacante para realizar ataques, inyecciones o modificar los privilegios de ejecución. En el peor de los casos, la deserialización insegura puede llevar a la ejecución de código remoto en el servidor.
Componentes con vulnerabilidades conocidas	Los componentes como las bibliotecas, los marcos de trabajo y otros módulos tienen el mismo privilegio de la aplicación. Si se usa un componente vulnerable, el ataque puede causar pérdida de datos o control de servidor. Las aplicaciones y las API que utilizan componentes que tienen alguna vulnerabilidad puede debilitar los sistemas de defensa y permitir diferentes ataques e impactos.
Registro y Monitoreo Insuficientes	El monitoreo inadecuado, junto con la falta de reacciones ante problemas de seguridad, permiten que un atacante mantenga los ataques a lo largo del tiempo y manipule o destruya los datos. Los estudios muestran el tiempo de descubrimiento de una brecha de seguridad suele ser de

	más de 200 días, y a menudo detectados por terceros en lugar de los dueños del sistema.

Fuente: (Owasp.org., 2017)

Tabla 3

Nivel de riesgo por cada vulnerabilidad de seguridad según OWASP.

Riesgos de Seguridad en Aplicaciones	**Explotabilidad**	**Prevalencia**	**Detectabilidad**	**Impacto**
Inyección	3	2	3	3
Pérdida de Autenticación	3	2	2	3
Exposición de datos sensibles	2	3	2	3
Entidades externas XML (XEE)	2	2	3	3
Pérdida de control de Acceso	2	2	2	3
Configuración de Seguridad Incorrecta	3	3	3	2
Secuencia de Comandos en Sitios Cruzados (XSS)	3	3	3	2

Deserialización Insegura	1	2	2	3
Componentes con vulnerabilidades conocidas	2	3	2	2
Registro y Monitoreo Insuficientes	2	3	1	2

Fuente: (Owasp.org., 2017)

Las características de los modelos SAMM y Microsoft-SDL fueron analizadas y seleccionadas con el criterio de dificultad de uso y tiempo consumido para realizar los procesos. Gracias a esto se obtuvieron la lista de características más aptas para el modelo de procesos propuesto.

Tabla 4

Selección de características de seguridad de los modelos SAMM y Microsoft SDL aptas para el modelo propuesto.

	Criterios	
Características	**Difícil de realizar**	**Consume mucho tiempo**
Estrategias y Métricas	NO	NO
Políticas y Cumplimiento	SI	SI
Educación y Orientación	SI	SI
Evaluación de Amenaza	SI	SI
Requisitos de Seguridad	NO	NO

Arquitectura de Seguridad	SI	SI
Revisión de Diseño	SI	SI
Revisión de Código	NO	NO
Pruebas de Seguridad	SI	SI
Administración de Vulnerabilidades	SI	SI
Fortalecimiento del Ambiente	SI	SI
Habilidad Operativa	SI	SI
Requisitos de Información	SI	SI
Requisitos de Seguridad	NO	NO
Umbrales de Calidad y Limites de Errores	SI	SI
Evaluación de los Riesgos de Seguridad y Privacidad	SI	SI
Requisitos de Diseño	NO	NO
Reducción de la superficie de ataques	SI	SI
Modelos de Riesgo	SI	SI
Usar herramientas aprobadas	NO	NO
Prohibir funciones no seguras	NO	NO
Análisis Estático	NO	NO
Análisis Dinámico	NO	NO
Pruebas de exploración de vulnerabilidades	SI	SI

mediante datos aleatorios		
Revisión de los modelos de riesgos y de la superficie de ataques	SI	SI
Plan de respuesta a incidentes	NO	NO
Revisión de seguridad final	NO	NO

Fuente: Elaboración propia

Las buenas prácticas para prevenir las vulnerabilidades de seguridad más recurrentes identificadas anteriormente, se encuentran descritas en las investigaciones de OWASP en su documento OWASP TOP 10 – 2017 Los diez riesgos más críticos en Aplicaciones Web.

Tabla 5

Resumen de buenas prácticas definidas por OWASP para evitar riesgos de seguridad más recurrentes en aplicaciones.

Riesgos de Seguridad en Aplicaciones	**Cómo prevenirlos**
Inyección	Para evitar la inyección SQL, es necesario separar y escapar los datos ingresados por el usuario de las consultas y los comandos. • La opción preferida es utilizar una API segura que evita por completo el uso del compilador y proporciona una interfaz parametrizada. Debe migrar y utilizar la herramienta de Mapeo Relacional de Objetos

	• Valide las entradas de datos en el servidor, utilizando una "lista blanca". Sin embargo, esta no es una protección completa porque muchas aplicaciones requieren el uso de caracteres especiales, como en campos de texto, API o aplicaciones móviles. • Use LIMIT y otros controles de SQL en las consultas para evitar la vulneración de varios registros en caso de una inserción de SQL.
Pérdida de Autenticación	• Implemente la autenticación multifactor para evitar ataques automatizados, ataques de fuerza bruta o reutilización de credenciales robadas. • Realice una verificación de contraseña débil. Cuando un usuario ingresa una nueva contraseña, se puede comparar con una lista de 10,000 contraseñas débiles. • Ajuste las políticas de longitud, complejidad y rotación de contraseñas y las políticas de contraseñas existentes. • Determinar o aumentar el tiempo de respuesta para cada intento fallido de inicio de sesión. Registre todos los bloqueos y notifique a los administradores cuando se detecten ataques de fuerza bruta.
Exposición de datos sensibles	Como mínimo, siga estos consejos y verifique las referencias: • Clasificación de los datos procesados, almacenados o transmitidos por el sistema. Identifique la información confidencial en función de las reglamentaciones, leyes o requisitos nacionales y de la empresa.

	• Implementar medidas de control adecuadas para cada categoría. • No almacene datos confidenciales innecesariamente o en su defecto elimínelos lo antes posible. • Utilice solo algoritmos y protocolos estándar sólidos e implemente una gestión de claves adecuada. • Deshabilitar el almacenamiento en caché de datos confidenciales.
Entidades externas XML (XEE)	La capacitación de los desarrolladores es esencial para identificar y mitigar las fallas XXE. Además, el bloqueo de XXE requiere: • Si es posible, utilice formatos de datos simples como JSON y evite la serialización de datos confidenciales. • Actualizar los procesadores y bibliotecas XML que utiliza la aplicación o la plataforma. Utilice el comprobador de dependencias. • Implemente la validación de entrada activa del lado del servidor ("lista blanca") y el filtrado para evitar que datos maliciosos entren en documentos XML, encabezados y nodos. • Verifique que la función de carga de archivos XML o XSL valide el XML entrante utilizando la validación XSD o similar.
Pérdida de control de Acceso	El control de acceso solo es efectivo cuando se implementa en el lado del servidor o en API sin servidor, donde el atacante no puede modificar el control de acceso o los metadatos.

	• A excepción de los recursos públicos, esta política debe ser denegar todo por defecto. • El modelo de control de acceso debe imponer la propiedad (propiedad) de los registros, en lugar de aceptar la capacidad de los usuarios para crear, leer, actualizar o eliminar cualquier registro. • Desactive la lista de directorios del servidor web y asegúrese de que los metadatos/archivos de origen y las copias de seguridad no estén en carpetas públicas. • Registrar los errores de control de acceso y notifique a los administradores cuando corresponda. • Los desarrolladores y el personal de control de calidad deben incluir pruebas de control de acceso en sus pruebas unitarias y de integración.
Configuración de Seguridad Incorrecta	Se deben seguir los procedimientos de instalación segura, que incluyen: • Un proceso de procesamiento reproducible para acelerar y facilitar el desempeño de otro entorno protegido. Los entornos de desarrollo, control de calidad y producción deben configurarse de forma idéntica, con credenciales diferentes para cada uno. Este proceso se puede automatizar para reducir el esfuerzo necesario para configurar cada nuevo entorno seguro. • Utilice una plataforma optimizada sin características, componentes, documentación o ejemplos innecesarios. Elimine o desinstale marcos de trabajo y características no utilizados.

Secuencia de Comandos en Sitios Cruzados (XSS)	Para evitar las solicitudes XSS es necesario mantener los datos que no son de confianza separados del contenido activo del navegador. • Codifique datos de solicitudes HTTP que no sean de confianza en campos de salida HTML que resuelvan XSS reflejados y XSS almacenados. • Aplicar cifrado contextual al editar el documento en el navegador del cliente para ayudar a prevenir DOM XSS.
Deserialización Insegura	El único patrón arquitectónico seguro es no aceptar objetos serializados de fuentes que no son de confianza o utilizar medios de serialización que solo permiten tipos de datos sin procesar. Si eso no es posible, considere uno de los siguientes: • En la deserialización y antes de crear objetos, imponga una verificación de tipo estricta, ya que el código a menudo tiene un conjunto de clases definidas. Se ha comprobado que esta técnica se puede ignorar, por lo que no se recomienda confiar únicamente en ella. • Guarde un registro de las excepciones y errores de deserialización, por ejemplo, cuando el tipo recibido no es el esperado o la deserialización produce algún tipo de error.
Componentes con vulnerabilidades conocidas	• Elimine dependencias, características, componentes, archivos y documentos innecesarios y no utilizados. • Obtenga componentes solo de fuentes oficiales utilizando canales seguros. Prefiere paquetes

	firmados para reducir el riesgo de usar versiones manipuladas malintencionadamente.
Registro y Monitoreo Insuficientes	En función de los riesgos de los datos almacenados o tratados por la aplicación: • Asegúrese de que todos los inicios de sesión del lado del servidor, el control de acceso y las fallas de validación de entrada de datos puedan registrarse para identificar cuentas sospechosas. Que el tiempo que los registros se mantengan activos sea lo suficientemente largo para permitir que se lleve a cabo cualquier análisis forense. • Asegúrese de que las transacciones de alto impacto tengan un registro de auditoría con controles de integridad para evitar la manipulación o la eliminación. • Asegúrese de que todas las transacciones de alto valor se auditen con verificación de integridad para detectar modificaciones o eliminaciones, como bases de datos con solo permisos de inserción o similares. • Configure un monitoreo y alerta efectivos para que la actividad sospechosa sea detectada y respondida dentro de un marco de tiempo aceptable.

Fuente: (Owasp.org., 2017)

Para mejorar el proceso de una organización que desarrolla aplicaciones y APIs, OWASP sugiere el Modelo de Garantía de la Madurez del Software (SAMM). Este modelo permite a las organizaciones exponer y poner en marcha estrategias para el software seguro, ajustado a los riesgos

específicos para su negocio y organización. Estas son las estrategias que plantea el modelo:

- Crear un plan estratégico integrado para la seguridad del software dentro de la organización.
- Determinar el valor relativo de los datos y bienes, y seleccionar la tolerancia al riesgo.
- Ajustar los gastos de seguridad a los indicadores de negocio significativos y al valor de los activos.

3.1.2. Integrar los métodos, estrategias y buenas prácticas en un modelo de procesos.

Figura 6 Resumen del modelo de procesos. Elaboración propia.

El siguiente modelo de procesos propuesto se irá desarrollando cada etapa de manera sucesiva, las cuales tiene como resultado un producto individual (P) el cual será utilizado como base para desarrollar la siguiente etapa del modelo y así continuamente hasta finalizar el desarrollo del software garantizando así la aplicación de seguridad en cada etapa del desarrollo.

3.1.2.1. Estrategias a implementar

Antes de aplicar el método propuesto para incluir seguridad en cada etapa del desarrollo de software es de gran importancia que la empresa tome ciertas acciones con respecto al manejo de la seguridad de información, que analice sus fortalezas y debilidades con respecto a la

seguridad y sus riesgos. Aquí se muestran una serie de estrategias que se implementarán a lo largo del tiempo, que si bien es cierto aplicarlas puede conllevar a un aumento de costos y tiempo en primera instancia, ayudará a que tanto la organización como sus trabajadores analicen y apliquen de manera eficiente este y otros modelos de procesos más complejos, ya que tendrán una base fuerte en cuanto a estrategias de seguridad refiere.

3.1.2.1.1. Crear un plan integrado para la seguridad de software dentro de la organización.

3.1.2.1.1.1. Tareas:

A. Evaluar el perfil global de riesgo del negocio.

Se efectúa una entrevista a los dueños del negocio y se establece un listado de los peores cuadros a lo largo de varios bienes de aplicación y datos. Luego se disponen y eligen los más importantes de acuerdo a la información reunida, seguido se proyecta una delineación de cada componente elegido y certificar los detalles de los causantes que contribuyen a los peores escenarios.

B. Establecer un plan de implementación para el programa de aseguramiento.

Atribuir una puntuación a cada práctica de 1, 2 o 3 según el objetivo concerniente según el éxito de cada mérito. En caso no se cumpliera ninguno se asigna una calificación de cero. Luego de entender bien el estado actual, se identifican las prácticas que serán perfeccionadas de acuerdo al perfil de riesgo. Luego, las acciones de mejora en el programa de aseguramiento deberán ejecutarse en tres hasta seis meses, teniendo una sesión de revisión cada tres meses para conocer el progreso del programa y hacer los ajustes necesarios al programa.

3.1.2.1.2. Determinar el valor relativo de los datos y bienes, y seleccionar la tolerancia al riesgo.

3.1.2.1.2.1. Tareas:

A. Clasificar datos y aplicaciones basados en riesgo de negocio.

Se debe constituir un sistema simple de categorización para interpretar el rango de riesgo por aplicación, crear pautas de evaluación de proyecto que asigne una categoría de riesgo a cada proyecto. Luego se debe evaluar la información reunida sobre cada aplicación y atribuir una categoría de riesgo a de acuerdo a los criterios de evaluación. Luego se debe plantear un proceso constante para la clasificación de riesgo de aplicaciones y bienes de información y mantener al día la información actual al menos 2 veces por año.

B. Plantear los objetivos de seguridad según cada clasificación.

El plan de implementación del programa de aseguramiento debe variar para considerar cada categoría de riesgo de las aplicaciones al hacer hincapié en prácticas particulares para cada categoría. En este procedimiento se debe tomar decisiones activas sobre qué objetivos específicos se desean de las aplicaciones en cada categoría de riesgo. Al optar por mantener aplicaciones de menor riesgo en niveles menores de desempeño en relación a las prácticas de seguridad, se reducen recursos a cambio de la aprobación de un riesgo mesurado.

A continuación, con esta base, la organización aplicará el modelo de procesos como principal estrategia de seguridad para el desarrollo de software, la cual garantizará la seguridad en el software desarrollado desde etapas tempranas, el cual se detallará a continuación.

3.1.2.1.3. FASE DE REQUERIMIENTOS

3.1.2.1.3.1. Requisitos de Seguridad

Definir las especificaciones de seguridad mínimas.

Producto (P1): Lista de requisitos de seguridad.

3.1.2.1.3.2. Controles de Roles y Privilegios

Definir los roles y privilegios de cada módulo a desarrollar.

Producto (P2): Documento de Roles y Privilegios por módulos.

Tabla 6

Ejemplo de la primera parte del Producto P2.

MODULOS DEL SOFTWARE		
ID	**MÓDULO (D)**	**DESCRIPCIÓN**
D1	Administración de Clientes	Permite agregar, modificar y eliminar a los clientes.
D2	Módulo de Ventas	Permite realizar ventas, boletas, facturas, notas de credito …
…	---	…
Dn	Módulo de Reportes	…

Fuente: Elaboración propia.

Tabla 7

Ejemplo de la segunda parte del Producto P2.

C: AGREGAR **R:** VISUALIZAR **U:** MODIFICAR **D:** ELIMINAR

MATRIZ DE PRIVILEGIOS Y ROLES																	
ID	**ROL**	**D1**				**D2**				**D3**				**D4**			
		C	R	U	D	C	R	U	D	C	R	U	D	C	R	U	D
1	Administrador	•	•	•	•	•	•	•	•								
2	Operador									•	•	•	•				
3	Auditor													•	•	•	•
4	Cajero									•	•	•	•				
5	…																

Fuente: Elaboración propia.

3.1.2.1.4. FASE DE ANÁLISIS Y DISEÑO

3.1.2.1.4.1. Requisitos de diseño

Describir de forma precisa y completa el uso de una característica o función.

Describir cómo se implementa de forma segura la característica o función.

Producto (P3): Informe sobre cómo se aplicará seguridad en los procesos.

Tabla 8

Ejemplo de Producto P3.

MODULOS DEL SOFTWARE			
ID	**MÓDULO**	**DESCRIPCIÓN**	**SEGURIDAD**
D1	Administración de Clientes	Permite agregar, modificación y eliminar a los clientes.	Controlar la autenticación, validar sesión, validar datos ingresados, log de modificación
D2	Módulo de Ventas	Permite realizar ventas, boletas, facturas, …	Controlar la autenticación, validar datos ingresados, logs
…	---	…	…
Dn	Módulo de Reportes	…	…

Fuente: Elaboración propia

Producto (P4): Informe sobre las medidas de seguridad a aplicar en las interfaces de cada módulo.

Tabla 9

Ejemplo de Producto P4.

	Validación de Entradas	**Validación de Salidas**	**Controles de Seguridad**	**Manejo de Documentos**	**Registro de Logs**
D1	Corresponde	Corresponde	Corresponde	-	Corresponde

D2	-	-	-	Corresponde	-
...					

Fuente: Elaboración propia

3.1.2.1.5. FASE DE IMPLEMENTACIÓN Y CODIFICACIÓN

3.1.2.1.5.1. Buenas Prácticas (Top Ten OWASP)

Se deben seguir las indicaciones sobre cómo evitar las vulnerabilidades más recurrentes, identificadas en puntos anteriores, utilizando una tabla de doble entrada donde se identifica que vulnerabilidades se deben evitar de acuerdo al módulo que se esté desarrollando.

Producto (P5): Informe con las vulnerabilidades a evitar según el módulo a desarrollar.

Tabla 10

Ejemplo de Producto P5.

Vulnerabilidades	**D1**	**D2**	**D3**	**…**
Inyección	Aplica	-	-	
Pérdida de Autenticación	-	Aplica	-	
Exposición de datos sensibles	Aplica	-	Aplica	
…				

Fuente: Elaboración propia

3.1.2.1.5.2. Análisis Estático

Se debe analizar el código fuente con un software automatizado diseñado para realizar análisis estático del código fuente desarrollado. Tener en cuenta el tiempo que consume realizar estos análisis para no disminuir la productividad. En el siguiente ejemplo

se usó el software Progpilot para el análisis estático de un módulo de Login.

```
>php progpilot.phar --configuration ./configuration.yml ./login/
{
    "source_name": [
        "$getRealIP_return"
    ],
    "source_line": [
        58
    ],
    "source_column": [
        1571
    ],
    "source_file": [
        "C:\\Users\\Slaytonw\\Desktop\\login\\login.php"
    ],
    "sink_name": "echo",
    "sink_line": 62,
    "sink_column": 1624,
    "sink_file": "C:\\Users\\Slaytonw\\Desktop\\login\\login.php",
    "vuln_name": "xss",
    "vuln_cwe": "CWE_79",
    "vuln_id": "cca1e92935a4d8797259a0e0301c895671e762fb795c1c7a4dc90a699b0c2de7",
    "vuln_type": "taint-style"
},
{
    "source_name": [
        "$_GET[\"mm\"]"
    ],
    "source_line": [
        112
    ],
    "source_column": [
        3423
    ],
    "source_file": [
        "C:\\Users\\Slaytonw\\Desktop\\login\\login.php"
    ],
    "sink_name": "echo",
    "sink_line": 112,
    "sink_column": 3423,
    "sink_file": "C:\\Users\\Slaytonw\\Desktop\\login\\login.php",
    "vuln_name": "xss",
    "vuln_cwe": "CWE_79",
    "vuln_id": "cc7d9c15b818b4edb69721984a7916d9adeae4d87282d5463b61ca049838f063",
    "vuln_type": "taint-style"
},
```

Figura 7 Resultado del análisis estático usando Progpilot. Elaboración propia.

En el ejemplo se puede observar como el software Progpilot encontró 2 vulnerabilidades de tipo XSS.

Luego de realizar el análisis estático, con una herramienta, se procederá a registrar la información obtenida en el documento generado anteriormente. De manera que quede registrado si alguna vulnerabilidad no ha sido subsanada durante el desarrollo.

Producto (P6): Informe basado en el anterior, agregando los datos obtenidos por la herramienta de análisis estático del software.

Tabla 11

Ejemplo de Producto P6.

Vulnerabilidades	**D1**	**D2**	**D3**	**…**
Inyección	Aplica	-	-	
Pérdida de Autenticación	-	Aplica	-	
Exposición de datos sensibles	Aplica	-	Aplica	
…				
Análisis Estático	**1/2**	**1/1**	**1/1**	**n/n**

Fuente: Elaboración propia

3.1.2.1.6. FASE DE PRUEBAS

3.1.2.1.6.1. Análisis Dinámico del Software

Se debe analizar el software con programas de análisis en tiempo real, con datos aleatorios y simulando ataques, para así encontrar cualquier vulnerabilidad aun existente. En el siguiente ejemplo se usó el software OWASP ZAP para el análisis dinámico de un módulo de Login.

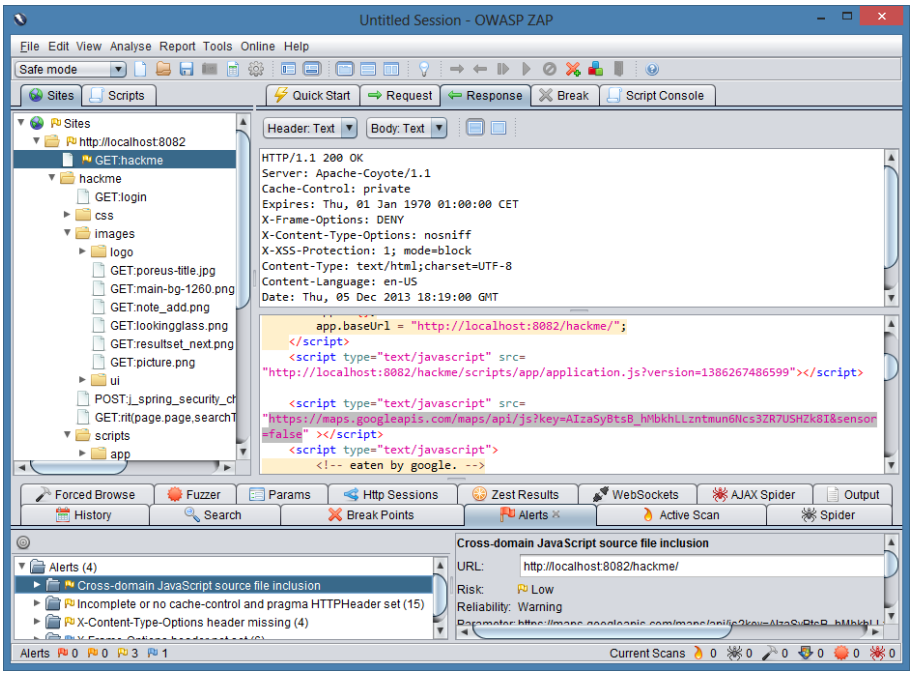


Figura 8 Resultado del análisis dinámico usando OWASP ZAP. Elaboración propia.

Luego de realizar el análisis dinámico, con una herramienta, se procederá a registrar la información obtenida en el documento generado anteriormente. De manera que quede registrado si alguna vulnerabilidad no ha sido subsanada en la fase de desarrollo.

Producto (P7): Informe basado en el anterior, agregando los datos obtenidos por la herramienta de análisis dinámico del software.

Tabla 12

Ejemplo de Producto P7.

Vulnerabilidades	**D1**	**D2**	**D3**	**…**
Inyección	Aplica	-	-	
Pérdida de Autenticación	-	Aplica	-	
Exposición de datos sensibles	Aplica	-	Aplica	
…				
Análisis Estático	**1/2**	**1/1**	**1/1**	**n/n**
Análisis Dinámico	**2/2**	**1/1**	**0/1**	**n/n**

Fuente: Elaboración propia

3.1.2.1.7. FASE DE LANZAMIENTO

3.1.2.1.7.1. Plan de respuesta a incidentes

Todo software lanzado con características de seguridad, debe tener un plan de respuesta a incidentes, el cual debe incluir los datos del personal a contactar en caso haya un incidente de seguridad, un contacto con capacidad de tomar decisiones y disponibilidad 24/7.

Producto (P8): Documento del Plan de Respuesta a Incidentes.

Tabla 13

Ejemplo de Producto P8.

Nombres y Apellidos	Cargo	Teléfono Principal
Contacto principal (Se registra el nombre de la persona encargada)	Jefe de Desarrollo	9xxxxxxxx (Se registra el teléfono)
Soporte (Se registra el nombre de la persona encargada)	Soporte 24/7	9xxxxxxxx (Se registra el teléfono)

Fuente: Elaboración propia

3.1.2.1.7.2. Revisión de Seguridad Final

Se realiza una inspección a todas las actividades de seguridad realizadas con un software antes de su lanzamiento. Con el único propósito de validar si el software puede o no ser lanzado. Teniendo como salida estos tres posibles resultados:

- Revisión de seguridad final superada. Se han corregido y mitigado todos los problemas de seguridad identificados al inicio.
- Revisión de seguridad final superada con excepciones. Se han corregido y mitigado todos los problemas de seguridad identificados al inicio, pero aún hay problemas de seguridad mínimos, los cuales serán solucionados en una siguiente versión.
- Revisión de seguridad final con remisión a una instancia superior. No se ha cumplido con los requisitos de seguridad identificados al inicio, el software no saldrá a producción. El equipo de desarrollo debe intentar solucionarlo antes de la fecha de lanzamiento o se comunicará a instancias superiores para tomar una decisión al respecto.

Producto (P9): Resultado de la revisión final en un documento. Se utiliza como base todos los productos realizados para ejecutar la revisión respectiva.

3.1.3. Validar el modelo propuesto.

Se dispuso ejecutar una evaluación con juicio de expertos que cuenten con experiencia en seguridad de software para que puedan estimar el modelo de procesos propuesto en este estudio. Para conseguir resultados positivos se decidió utilizar el método Delphi.

En la primera etapa del método se establecieron los elementos a consultar a los expertos, para esto se preparó un resumen de la propuesta, destacando las principales características del modelo de procesos.

En la segunda etapa de la metodología se distinguió y optó a los expertos que intervendrían en la evaluación de la propuesta de modelo de procesos, y fueron los siguientes:

- Ing. Ronald Ramírez Blanco
- Ing. Gino Begazo
- Ing. Christian Taipe Sedano

De acuerdo al perfil profesional de cada evaluador, se puede corroborar que los expertos tienen experiencia en seguridad de software y a la vez están certificados en el área de estudio. Luego de comunicarse con cada uno se procedió a enviarles a través de email, el resumen del modelo propuesto, además del cuestanario con 10 preguntas. Gracias a la adaptabilidad del método Delphi, se pudo realizar todo el proceso a través de email, y así conseguir la validación por juicio de expertos.

La primera fase de respuestas de cada evaluador se muestra a continuación:

Tabla 14

Respuestas de evaluadores en la primera iteración.

	Evaluador 1		**Evaluador 2**		**Evaluador 3**	
Pregunta		**Comentario**		**Comentario**		**Comentario**
Pregunta 1.- Según el modelo de	**SI**	Tiene las Fases básicas y esenciales que todo aplicativo	**SI**	Considero que sí ya que el modelo no es complejo de	**SI**	A pesar de tener fases de implementaci

procesos propuesto. **¿Considera usted que el modelo propuesto es viable para pequeñas empresas?**		debe contar para un proceso de construcción de Software		entender, no creo que dependa del tamaño del equipo de desarrollo ya que las actividades lo pueden realizar un programador promedio como avanzado y las actividades pueden ser desarrolladas por uno o varios integrantes de un equipo de desarrollo. Muchas veces no se implementa ya que no hay un modelo o una estructura de trabajo a seguir, contando con un modelo que no implica mayor complejidad sólo aumentar más actividades		ón que pueden ser bloqueantes y tardar más tiempo de lo previsto, teniendo un flujo exitoso se puede considerar viable.

				con respecto a la seguridad que beneficiará a corto o largo plazo el éxito de un proyecto.		
Pregunta 2.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto es de bajo coste de implementación?**	**SI**	Existen herramientas y procedimientos de libre acceso, los cuales coadyuvan a cualquier sistema a implementarse.	**SI**	No considero de bajo costo a primera vista, ya que eso dependerá del alcance del proyecto a realizar considero que de aplicar este modelo aumentaría del 10% al 30% el tiempo de desarrollo y por tal un aumento en presupuesto. Pero el beneficio se verá reflejado luego de la publicación o en etapa de producción reduciendo actividades de testeo o	**NO**	Dentro del modelo no especifica si el software usado es libre, además al ser un modelo por capas consecutivas podría demandar tiempo de implementación.

				inconvenientes que puedan surgir en cuanto a seguridad, reduciendo costos post producción en actividades de mantenimiento correctivas.		
Pregunta 3.- Según el modelo de procesos propuesto. **¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software**	**NO**	No se tiene especificado el análisis de Costo-Beneficio y tiempos	**SI**	Considero que sí ya que al usar estas herramientas se reduce tiempo en actividades de desarrollo o mantenimiento del software. Y económicamente ya que reducirá en el futuro presupuesto en mantenimientos correctivos y de forma indirecta reducción de daños causados por temas de seguridad.	**SI**	Considerando que el modelo se aplica desde el inicio del desarrollo y se ha estimado un tiempo para evitar y corregir los errores ayuda a reducir tiempo y dinero.

automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?						
Pregunta 4.- Según el modelo de procesos propuesto. **¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más**	**NO**	Solo cumple lo básico y necesario, no se están considerando puntos como entrenamiento en seguridad y pruebas de Penetración Testing.	**SI**	Considero que sí desde el punto de vista general, pero eso dependerá del proyecto de desarrollo en cuestión teniendo en cuenta el tipo de información que se guarda si es muy importante o sensible, la cantidad de usuarios que acceden al software, si es interna o pública, si es más propensa a ser atacada, etc. Según el	**SI**	Para una pequeña empresa con un software limitado si considero que sea un modelo suficiente para reducir las vulnerabilidades más recurrentes. Además, debe considerarse el entorno en el que se desplegara el software sea el caso interno o externo.

recurrentes?				nivel de seguridad considero que se agreguen más actividades, pero no reducir.		
Pregunta 5.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?**	**NO**	Aun así, considero que para empresas más grandes se puede mejorar en base a lo que se ha propuesto y elaborado, se puede desplegar y añadir actividades.	**NO**	Pienso que son los básicos e importantes que se deberían implementar.	**NO**	Dado el modelo y la planificación del mismo lo considero suficiente.
Pregunta 6.- Según el modelo de procesos propuesto.	**SI**	Contiene las pautas básicas y esenciales del SDLC-S, pero considero que se debería	**SI**	Considero que sí, la mayoría de proyectos de desarrollo solo lo aplica en la etapa de	**SI**	Si ya que las fases del modelo están detalladas y definen su aplicación.

¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?		profundizar más en las etapas de implementación y lanzamiento.		requerimientos, pero no en la etapa de análisis y diseño o en la implementación.		
Pregunta 7.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la**		- Fase de Requerimientos - Fase de Análisis y Diseño Todas las Fases son importantes en su magnitud, sin embargo, las mejores prácticas recomiendan que las 1eras fases son importantes para		- Fase de Pruebas Pienso que en la fase de pruebas se debe dar más importancia, ya que es la fase crítica de todo proyecto de software.		- Fase de Análisis y Diseño - Fase de Implementación y Codificación Son las fases que considero clave para llevar a cabo un buen análisis y posterior revisión de

seguridad ?		posteriormente no acarrear costos de diseño, mano de obra y redefinición de tiempos, es por ello, que las fases de requerimiento y análisis y diseño son las que deben ser mejor trabajadas.				bugs de seguridad.
Pregunta 8.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo**	**SI**	Contiene las Fases primarías, solo habría considerar la última etapa , donde se incluiría el mantenimiento de SW	**SI**	Pienso que sí la mayoría de metodologías de desarrollo de software manejan tres fases principales requerimientos, análisis, desarrollo, pruebas y lanzamiento, según la metodología en cualquier de las etapas hay	**NO**	Debido a que las metodologías de desarrollo son muy diversas y el modelo planteado es por fases y aplicado al final de la implementación del producto, esto dificulta la adaptación del modelo a metodologías

de software en cada una de sus fases?				actividades a desarrollar.		más actuales.
Pregunta 9.- Según el modelo de procesos propuesto. **¿Considera a usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?**	**NO**	Se tendría que entrar un poco más a detalle para otorgar una opinión a los productos resultantes.	**SI**	Si, ya que los productos entregados serán como un mapa para el equipo de desarrollo de las acciones realizadas en cada módulo con respecto a la seguridad del software.	**SI**	Solo si es aplicado a determinados productos según una metodología de desarrollo implementada en fases.
Pregunta 10.- Según el modelo de procesos propuesto. **¿Consider**	**NO**	El Modelo Propuesto contiene actividades básicas y esenciales sobre las	**SI**	Considero que sí, ya que si se realizan todas las actividades se aseguran tener un software que	**SI**	Solo si es aplicado a determinados productos desarrollado bajo metodologías

a usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?		características de seguridad, pero podría profundizarse en las fases de Implementación y lanzamiento, sin embargo, no contiene una metodología ágil de desarrollo y considero que esto podría mejorar		reduzca los inconvenientes con respecto a seguridad, que muchas veces se deja de lado, pero implementando este modelo también se reducirá a corto o largo plazo actividades correctivas.		realizadas por fases.

Fuente: Elaboración propia

Luego de la primera reiteración se afianzaron las respuestas y se les remitió dicha tabla a cada evaluador, prosiguiendo aún en el anonimato, para que los evaluadores analicen la consistecia de las respuestas proporcionadas. Las respuestas en la segunda reiteración de los otros evaluadores sólo permitieron confirmar o meditarr sobre las respuestas dadas en la primera reiiteración. Los evaluadores sostuvieron sus respuestas, como se muestra en la siguiente tabla.

Tabla 15

Respuesta del evaluador 01 en la segunda iteración

	Evaluador 1		**Evaluador 2**		**Evaluador 3**		**Segunda opinión**
Pregunta		**Comentario**		**Comentario**		**Comentario**	

Pregunta 1.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto es viable para pequeñas empresas?**	**SI**	Tiene las Fases básicas y esenciales que todo aplicativo debe contar para un proceso de construcción de Software	**SI**	Considero que sí ya que el modelo no es complejo de entender, no creo que dependa del tamaño del equipo de desarrollo ya que las actividades lo pueden realizar un programador promedio como avanzado y las actividades pueden ser desarrolladas por uno o varios integrantes de un equipo de desarrollo. Muchas veces no se implementa	**SI**	A pesar de tener fases de implementación que pueden ser bloqueantes y tardar más tiempo de lo previsto, teniendo un flujo exitoso se puede considerar viable.	Conservo mi opinión.

				ya que no hay un modelo o una estructura de trabajo a seguir, contando con un modelo que no implica mayor complejidad sólo aumentar más actividades con respecto a la seguridad que beneficiará a corto o largo plazo el éxito de un proyecto.			
Pregunta 2.- Según el modelo de procesos propuesto. **¿Consider**	**SI**	Existen herramientas y procedimientos de libre acceso, los	**SI**	No considero de bajo costo a primera vista, ya que eso	**NO**	Dentro del modelo no especifica si el software usado es libre, además al	Conservo mi opinión.

a usted que el modelo propuesto es de bajo coste de implementación?		cuales coadyuvan a cualquier sistema a implementarse.		dependerá del alcance del proyecto a realizar considero que de aplicar este modelo aumentaría del 10% al 30% el tiempo de desarrollo y por tal un aumento en presupuesto. Pero el beneficio se verá reflejado luego de la publicación o en etapa de producción reduciendo actividades de testeo o inconvenientes que puedan surgir en cuanto a		ser un modelo por capas consecutivas podría demandar tiempo de implementación.	

				seguridad, reduciendo costos post producción en actividades de mantenimiento correctivas.			
Pregunta 3.- Según el modelo de procesos propuesto. **¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software**	**NO**	No se tiene especificado el análisis de Costo-Beneficio y tiempos	**SI**	Considero que sí ya que al usar estas herramientas se reduce tiempo en actividades de desarrollo o mantenimiento del software. Y económicamente ya que reducirá en el futuro presupuesto en mantenimientos correctivos y de forma	**SI**	Considerando que el modelo se aplica desde el inicio del desarrollo y se ha estimado un tiempo para evitar y corregir los errores ayuda a reducir tiempo y dinero.	Conservo mi opinión.

automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?				indirecta reducción de daños causados por temas de seguridad.			
Pregunta 4.- Según el modelo de procesos propuesto. **¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más**	**NO**	Solo cumple lo básico y necesario, no se están considerando puntos como entrenamiento en seguridad y pruebas de Penetración Testing.	**SI**	Considero que sí desde el punto de vista general, pero eso dependerá del proyecto de desarrollo en cuestión teniendo en cuenta el tipo de información que se guarda si es muy importante o sensible, la cantidad de usuarios que	**SI**	Para una pequeña empresa con un software limitado si considero que sea un modelo suficiente para reducir las vulnerabilidades más recurrentes. Además, debe considerarse el entorno en el que se desplegara el software sea el caso	Conservo mi opinión.

recurrentes?				acceden al software, si es interna o pública, si es más propensa a ser atacada, etc. Según el nivel de seguridad considero que se agreguen más actividades, pero no reducir.		interno o externo.	
Pregunta 5.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática retiraría o agregaría algún**	**NO**	Aun así, considero que para empresas más grandes se puede mejorar en base a lo que se ha propuesto y elaborado, se puede desplegar y añadir	**NO**	Pienso que son los básicos e importantes que se deberían implementar.	**NO**	Dado el modelo y la planificación del mismo lo considero suficiente.	Conservo mi opinión.

proceso al modelo propuesto?		actividades.					
Pregunta 6.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?**	**SI**	Contiene las pautas básicas y esenciales del SDLC-S, pero considero que se debería profundizar más en las etapas de implementación y lanzamiento.	**SI**	Considero que sí, la mayoría de proyectos de desarrollo solo lo aplica en la etapa de requerimientos, pero no en la etapa de análisis y diseño o en la implementación.	**SI**	Si ya que las fases del modelo están detalladas y definen su aplicación.	Conservo mi opinión.
Pregunta 7.- Según el modelo de procesos propuesto. **¿Según su**		- Fase de Requerimientos - Fase de Análisis y Diseño		- Fase de Pruebas Pienso que en la fase de pruebas se debe dar más		- Fase de Análisis y Diseño - Fase de Implementación y Codificación	Conservo mi opinión.

experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?		Todas las Fases son importantes en su magnitud, sin embargo, las mejores prácticas recomiendan que las 1eras fases son importantes para posteriormente no acarrear costos de diseño, mano de obra y redefinición de tiempos, es por ello, que las fases de requerimiento y análisis y diseño son		importancia, ya que es la fase crítica de todo proyecto de software.		Son las fases que considero clave para llevar a cabo un buen análisis y posterior revisión de bugs de seguridad.	

		las que deben ser mejor trabajadas.					
Pregunta 8.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?**	**SI**	Contiene las Fases primarías, solo habría considerar la última etapa , donde se incluiría el mantenimiento de SW	**SI**	Pienso que sí la mayoría de metodologías de desarrollo de software manejan tres fases principales requerimientos, análisis, desarrollo, pruebas y lanzamiento , según la metodología en cualquier de las etapas hay actividades a desarrollar.	**NO**	Debido a que las metodologías de desarrollo son muy diversas y el modelo planteado es por fases y aplicado al final de la implementación del producto, esto dificulta la adaptación del modelo a metodologías más actuales.	Conservo mi opinión.
Pregunta 9.- Según el modelo de procesos	**NO**	Se tendría que entrar un poco más a detalle	**SI**	Si, ya que los productos entregados serán como	**SI**	Solo si es aplicado a determinados productos según una	Conservo mi opinión.

propuesto. **¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?**		para otorgar una opinión a los productos resultantes.		un mapa para el equipo de desarrollo de las acciones realizadas en cada módulo con respecto a la seguridad del software.		metodología de desarrollo implementada en fases.	
Pregunta 10.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrolla**	**NO**	El Modelo Propuesto contiene actividades básicas y esenciales sobre las características de seguridad, pero podría profundizarse en las fases de Implementación y lanzamient	**SI**	Considero que sí, ya que si se realizan todas las actividades se aseguran tener un software que reduzca los inconvenientes con respecto a seguridad, que muchas veces se deja de	**SI**	Solo si es aplicado a determinados productos desarrollado bajo metodologías realizadas por fases.	Conservo mi opinión.

r software con características de seguridad ?		o, sin embargo, no contiene una metodología ágil de desarrollo y considero que esto podría mejorar		lado, pero implementando este modelo también se reducirá a corto o largo plazo actividades correctivas.			

Fuente: Elaboración propia

Tabla 16

Respuesta del evaluador 02 en la segunda iteración

	Evaluador 1		**Evaluador 2**		**Evaluador 3**		**Segunda opinión**
Pregunta		**Comentario**		**Comentario**		**Comentario**	
Pregunta 1.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto**	**SI**	Tiene las Fases básicas y esenciales que todo aplicativo debe contar para un proceso de construcci	**SI**	Considero que sí ya que el modelo no es complejo de entender, no creo que dependa del tamaño del equipo de	**SI**	A pesar de tener fases de implementación que pueden ser bloqueantes y tardar más tiempo de lo previsto, teniendo un	Conservo mi opinión.

es viable para pequeñas empresas ?		ón de Software		desarrollo ya que las actividades lo pueden realizar un programador promedio como avanzado y las actividades pueden ser desarrolladas por uno o varios integrantes de un equipo de desarrollo. Muchas veces no se implementa ya que no hay un modelo o una estructura de trabajo a seguir, contando con un modelo que no implica		flujo exitoso se puede considerar viable.	

				mayor complejidad sólo aumentar más actividades con respecto a la seguridad que beneficiará a corto o largo plazo el éxito de un proyecto.			
Pregunta 2.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto es de bajo coste de implementación?**	**SI**	Existen herramientas y procedimientos de libre acceso, los cuales coadyuvan a cualquier sistema a implementarse.	**SI**	No considero de bajo costo a primera vista, ya que eso dependerá del alcance del proyecto a realizar considero que de aplicar este modelo aumentaría del 10% al 30% el	**NO**	Dentro del modelo no especifica si el software usado es libre, además al ser un modelo por capas consecutivas podría demandar tiempo de implementación.	Conservo mi opinión.

				tiempo de desarrollo y por tal un aumento en presupuesto. Pero el beneficio se verá reflejado luego de la publicación o en etapa de producción reduciendo actividades de testeo o inconvenientes que puedan surgir en cuanto a seguridad, reduciendo costos post producción en actividades de mantenimiento correctivas.			

Pregunta 3.- Según el modelo de procesos propuesto. **¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?**	**NO**	No se tiene especificado el análisis de Costo-Beneficio y tiempos	**SI**	Considero que sí ya que al usar estas herramientas se reduce tiempo en actividades de desarrollo o mantenimiento del software. Y económicamente ya que reducirá en el futuro presupuesto en mantenimientos correctivos y de forma indirecta reducción de daños causados por temas de seguridad.	**SI**	Considerando que el modelo se aplica desde el inicio del desarrollo y se ha estimado un tiempo para evitar y corregir los errores ayuda a reducir tiempo y dinero.	Conservo mi opinión.

Pregunta 4.- Según el modelo de procesos propuesto. **¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?**	**NO**	Solo cumple lo básico y necesario, no se están considerando puntos como entrenamiento en seguridad y pruebas de Penetración Testing.	**SI**	Considero que sí desde el punto de vista general, pero eso dependerá del proyecto de desarrollo en cuestión teniendo en cuenta el tipo de información que se guarda si es muy importante o sensible, la cantidad de usuarios que acceden al software, si es interna o pública, si es más propensa a ser atacada, etc. Según el nivel de	**SI**	Para una pequeña empresa con un software limitado si considero que sea un modelo suficiente para reducir las vulnerabilidades más recurrentes. Además, debe considerarse el entorno en el que se desplegara el software sea el caso interno o externo.	Conservo mi opinión.

				seguridad considero que se agreguen más actividades, pero no reducir.			
Pregunta 5.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?**	**NO**	Aun así, considero que para empresas más grandes se puede mejorar en base a lo que se ha propuesto y elaborado, se puede desplegar y añadir actividades.	**NO**	Pienso que son los básicos e importantes que se deberían implementar.	**NO**	Dado el modelo y la planificación del mismo lo considero suficiente.	Conservo mi opinión.
Pregunta 6.- Según el modelo de procesos	**SI**	Contiene las pautas básicas y esenciales del SDLC-	**SI**	Considero que sí, la mayoría de proyectos de	**SI**	Si ya que las fases del modelo están detalladas y	Conservo mi opinión.

propuesto. **¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?**		S, pero considero que se debería profundizar más en las etapas de implementación y lanzamiento.		desarrollo solo lo aplica en la etapa de requerimientos, pero no en la etapa de análisis y diseño o en la implementación.		definen su aplicación.	
Pregunta 7.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática en cuál fase se debe darle más importanc**		- Fase de Requerimientos - Fase de Análisis y Diseño Todas las Fases son importantes en su magnitud, sin embargo, las mejores prácticas		- Fase de Pruebas Pienso que en la fase de pruebas se debe dar más importancia, ya que es la fase crítica de todo proyecto de software.		- Fase de Análisis y Diseño - Fase de Implementación y Codificación Son las fases que considero clave para llevar a cabo un buen análisis y posterior	Conservo mi opinión.

ia a la seguridad ?		recomiendan que las 1eras fases son importantes para posteriormente no acarrear costos de diseño, mano de obra y redefinición de tiempos, es por ello, que las fases de requerimiento y análisis y diseño son las que deben ser mejor trabajadas.				revisión de bugs de seguridad.	
Pregunta 8.- Según el modelo de procesos propuesto.	**SI**	Contiene las Fases primarías, solo habría considerar la última	**SI**	Pienso que sí la mayoría de metodologías de desarrollo	**NO**	Debido a que las metodologías de desarrollo son muy	Conservo mi opinión.

¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?		etapa , donde se incluiría el mantenimiento de SW		de software manejan tres fases principales requerimientos, análisis, desarrollo, pruebas y lanzamiento, según la metodología en cualquier de las etapas hay actividades a desarrollar.		diversas y el modelo planteado es por fases y aplicado al final de la implementación del producto, esto dificulta la adaptación del modelo a metodologías más actuales.	
Pregunta 9.- Según el modelo de procesos propuesto. **¿Considera usted que los productos resultantes del modelo propuesto son de**	**NO**	Se tendría que entrar un poco más a detalle para otorgar una opinión a los productos resultantes.	**SI**	Si, ya que los productos entregados serán como un mapa para el equipo de desarrollo de las acciones realizadas en cada módulo con respecto a	**SI**	Solo si es aplicado a determinados productos según una metodología de desarrollo implementada en fases.	Conservo mi opinión.

utilidad para el desarrollo del software?				la seguridad del software.			
Pregunta 10.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?**	**NO**	El Modelo Propuesto contiene actividades básicas y esenciales sobre las características de seguridad, pero podría profundizarse en las fases de Implementación y lanzamiento, sin embargo, no contiene una metodología ágil de desarrollo y considero	**SI**	Considero que sí, ya que si se realizan todas las actividades se aseguran tener un software que reduzca los inconvenientes con respecto a seguridad, que muchas veces se deja de lado, pero implementando este modelo también se reducirá a corto o largo plazo actividades correctivas.	**SI**	Solo si es aplicado a determinados productos desarrollado bajo metodologías realizadas por fases.	Conservo mi opinión.

		que esto podría mejorar					

Fuente: Elaboración propia

Tabla 17

Respuesta del evaluador 03 en la segunda iteración

	Evaluador 1		**Evaluador 2**		**Evaluador 3**		**Segunda opinión**
Pregunta		**Comentario**		**Comentario**		**Comentario**	
Pregunta 1.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto es viable para pequeñas empresas?**	**SI**	Tiene las Fases básicas y esenciales que todo aplicativo debe contar para un proceso de construcción de Software	**SI**	Considero que sí ya que el modelo no es complejo de entender, no creo que dependa del tamaño del equipo de desarrollo ya que las actividades lo pueden realizar un programador promedio como avanzado y	**SI**	A pesar de tener fases de implementación que pueden ser bloqueantes y tardar más tiempo de lo previsto, teniendo un flujo exitoso se puede considerar viable.	Conservo mi opinión.

				las actividades pueden ser desarrolladas por uno o varios integrantes de un equipo de desarrollo. Muchas veces no se implementa ya que no hay un modelo o una estructura de trabajo a seguir, contando con un modelo que no implica mayor complejidad sólo aumentar más actividades con respecto a la seguridad			

				que beneficiará a corto o largo plazo el éxito de un proyecto.			
Pregunta 2.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto es de bajo coste de implementación?**	**SI**	Existen herramientas y procedimientos de libre acceso, los cuales coadyuvan a cualquier sistema a implementarse.	**SI**	No considero de bajo costo a primera vista, ya que eso dependerá del alcance del proyecto a realizar considero que de aplicar este modelo aumentaría del 10% al 30% el tiempo de desarrollo y por tal un aumento en presupuesto. Pero el beneficio se verá reflejado	**NO**	Dentro del modelo no especifica si el software usado es libre, además al ser un modelo por capas consecutivas podría demandar tiempo de implementación.	Conservo mi opinión.

				luego de la publicación o en etapa de producción reduciendo actividades de testeo o inconvenientes que puedan surgir en cuanto a seguridad, reduciendo costos post producción en actividades de mantenimiento correctivas.			
Pregunta 3.- Según el modelo de procesos propuesto. **¿Considera usted que en las fases de**	**NO**	No se tiene especificado el análisis de Costo-Beneficio y tiempos	**SI**	Considero que sí ya que al usar estas herramientas se reduce tiempo en actividades de desarrollo o	**SI**	Consideran do que el modelo se aplica desde el inicio del desarrollo y se ha estimado un tiempo para	Conservo mi opinión.

codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?				mantenimiento del software. Y económicamente ya que reducirá en el futuro presupuesto en mantenimientos correctivos y de forma indirecta reducción de daños causados por temas de seguridad.		evitar y corregir los errores ayuda a reducir tiempo y dinero.	
Pregunta 4.- Según el modelo de procesos propuesto. **¿Considera usted que los procesos definidos**	**NO**	Solo cumple lo básico y necesario, no se están consideran do puntos como entrenamiento en	**SI**	Considero que sí desde el punto de vista general, pero eso dependerá del proyecto de desarrollo	**SI**	Para una pequeña empresa con un software limitado si considero que sea un modelo suficiente para reducir	Conservo mi opinión.

en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?		seguridad y pruebas de Penetración Testing.		en cuestión teniendo en cuenta el tipo de información que se guarda si es muy importante o sensible, la cantidad de usuarios que acceden al software, si es interna o pública, si es más propensa a ser atacada, etc. Según el nivel de seguridad considero que se agreguen más actividades, pero no reducir.		las vulnerabilidades más recurrentes. Además, debe considerarse el entorno en el que se desplegara el software sea el caso interno o externo.	
Pregunta 5.- Según el modelo	**NO**	Aun así, considero que para	**NO**	Pienso que son los básicos e	**NO**	Dado el modelo y la planificación	Conservo mi opinión.

de procesos propuesto. **¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?**		empresas más grandes se puede mejorar en base a lo que se ha propuesto y elaborado, se puede desplegar y añadir actividades.		importantes que se deberían implementar.		del mismo lo considero suficiente.	
Pregunta 6.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto asegura características de seguridad en cada**	**SI**	Contiene las pautas básicas y esenciales del SDLC-S, pero considero que se debería profundizar más en las etapas de implementación y lanzamiento.	**SI**	Considero que sí, la mayoría de proyectos de desarrollo solo lo aplica en la etapa de requerimientos, pero no en la etapa de análisis y diseño o en la	**SI**	Si ya que las fases del modelo están detalladas y definen su aplicación.	Conservo mi opinión.

una de las etapas del desarrollo de software?				implementación.			
Pregunta 7.- Según el modelo de procesos propuesto. **¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?**		- Fase de Requerimientos - Fase de Análisis y Diseño Todas las Fases son importantes en su magnitud, sin embargo, las mejores prácticas recomiendan que las 1eras fases son importantes para posteriormente no acarrear costos de diseño,		- Fase de Pruebas Pienso que en la fase de pruebas se debe dar más importancia, ya que es la fase crítica de todo proyecto de software.		- Fase de Análisis y Diseño - Fase de Implementación y Codificación Son las fases que considero clave para llevar a cabo un buen análisis y posterior revisión de bugs de seguridad.	Conservo mi opinión.

		mano de obra y redefinición de tiempos, es por ello, que las fases de requerimiento y análisis y diseño son las que deben ser mejor trabajadas.					
Pregunta 8.- Según el modelo de procesos propuesto. **¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de**	**SI**	Contiene las Fases primarías, solo habría considerar la última etapa , donde se incluiría el mantenimiento de SW	**SI**	Pienso que sí la mayoría de metodologías de desarrollo de software manejan tres fases principales requerimientos, análisis, desarrollo, pruebas y lanzamiento , según la metodología	**NO**	Debido a que las metodologías de desarrollo son muy diversas y el modelo planteado es por fases y aplicado al final de la implementación del producto, esto dificulta la	Conservo mi opinión.

desarrollo de software en cada una de sus fases?				en cualquier de las etapas hay actividades a desarrollar.		adaptación del modelo a metodología s más actuales.	
Pregunta 9.- Según el modelo de procesos propuesto. **¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?**	**NO**	Se tendría que entrar un poco más a detalle para otorgar una opinión a los productos resultantes.	**SI**	Si, ya que los productos entregados serán como un mapa para el equipo de desarrollo de las acciones realizadas en cada módulo con respecto a la seguridad del software.	**SI**	Solo si es aplicado a determinado s productos según una metodología de desarrollo implementa da en fases.	Conservo mi opinión.
Pregunta 10.- Según el modelo de procesos propuesto.	**NO**	El Modelo Propuesto contiene actividades básicas y esenciales	**SI**	Considero que sí, ya que si se realizan todas las actividades	**SI**	Solo si es aplicado a determinado s productos desarrollado bajo	Conservo mi opinión.

¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?		sobre las características de seguridad, pero podría profundizarse en las fases de Implementación y lanzamiento, sin embargo, no contiene una metodología ágil de desarrollo y considero que esto podría mejorar		se aseguran tener un software que reduzca los inconvenientes con respecto a seguridad, que muchas veces se deja de lado, pero implementando este modelo también se reducirá a corto o largo plazo actividades correctivas.		metodologías realizadas por fases.	

Fuente: Elaboración propia

En base a al juicio de expertos se realizaron las afirmaciones descritas anteriormente en análisis y discusión de resultados.

3.2. Consideraciones finales

Las vulnerabilidades más recurrentes en el software cambian constantemente en el tiempo, debido a que la tecnología siempre va cambiando, suelen aparecer nuevas vulnerabilidades, es por eso que se

debe estar siempre al tanto de estos cambios, para asi poder mantener los softwares seguros contra cualquier ataque malicioso.

Actualmente los modelos de procesos más conocidos e importantes son OpenSAMM y Microsoft-SDL los cuales proveen de muchos procesos y métodos a las empresas desarrolladoras de software, para que apliquen seguridad a sus procesos de desarrollo, pero muchos de esos métodos no son aplicables en pequeñas empresas de nuestra realidad, debido al gran consumo de tiempo, dinero y dificultad de aprendizaje. Es por eso que una selección de estos, los cuales sean relevantes y fáciles de usar en nuestra realidad, es la mejor forma de dar un paso hacia la seguridad en el desarrollo de software.

El desarrollo del modelo de procesos para integrar seguridad en el desarrollo del software, basado en una selección de métodos y estrategias de los modelos OpenSAMM y Microsoft-SDL, brinda una guía fácil de entender y funcional para los equipos de desarrollo de nuestra realidad, además que con el paso del tiempo puede servir como punto de partida para aplicar un modelo más robusto en la empresa.

Existen muchos estudios sobre las vulnerabilidades de seguridad en los softwares, cada uno orientado a un lenguaje de programación específico, en esta investigación se tuvo como punto principal el desarrollo web al momento de identificarlas.

El criterio utilizado en esta investigación para seleccionar los métodos y estrategias de los modelos existentes para la seguridad de software fue en base a la facilidad de uso y el consumo de tiempo, los cuales están orientados a equipos de desarrollo con bajo conocimiento en seguridad de software, para que de esta forma sea de fácil uso.

El modelo de procesos desarrollado en esta investigación tiene ciertas fases que incluyen el uso de herramientas externas, las cuales se seleccionan dependiendo de las vulnerabilidades a evaluar y el lenguaje desarrollado, en este caso se utilizó Progpilot para las pruebas estáticas y OWASP ZAP para las pruebas dinámicas.

REFERENCIAS

Baojiang, C., Baolian, L., & Tingting, H. (2014). Reverse analysis method of static XSS defect detection technique based on database query language. *9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.* Obtenido de https://doi.org/10.1109/3PGCIC.2014.99

Beck, K. (2004). *Extreme Programming Explained: Embrace Change* (Segunda ed.). Estados Unidos: Addison-Wesley.

Charette, R. (2004). *Why software fails* (Vol. 42). IEEE Spectrum.

Choi, H., Hong, S., Cho, S., & Kim, Y. G. (2018). HXD: Hybrid XSS detection by using a headless browser. Proceedings of the 2017. *4th International Conference on Computer Applications and Information Processing Technology.* Obtenido de https://doi.org/10.1109/CAIPT.2017.8320672

Deming, E. (1994). *La nueva economía para la industria, el gobierno y la educación.* Estados Unidos: The W. Edwards Deming Institute.

Díaz, R., Cerro, Y., & Proenza, P. (2014). *Sistema para la Gestión de la Información de Seguridad Informática en la Universidad de Ciencias Médicas de Holguín.* Cuba: Ciencias Holguín.

Frijns, P., Bierwolf, R., & Zijderhand, T. (2019). *Reframing Security in Contemporary Software Development Life Cycle.* IEEE International Conference on Technology Management, Operations and Decisions. Obtenido de https://doi.org/10.1109/ITMC.2018.8691277

Gartner. (2019). *Gartner Says Global IT Spending to Grow 1.1 Percent in 2019.*

Guo, X., Jin, S., & Zhang, Y. (2015)). XSS Vulnerability Detection Using Optimized Attack Vector Repertory. Proceedings - 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover. *CyberC 2015.* Obtenido de https://doi.org/10.1109/CyberC.2015.50

Highsmith, J. (2002). *Agile Software Development Ecosystems* (Vol. XIII). Boston: Addison-Wesley Professional.

Ivar, J., Grady, B., & James, R. (2000). *El proceso Unificado de Desarrollo de Software.* Addison Wesley.

Kaisler, S. (2005). *Software Paradigms.* New Jersey: John Wiley & Sons.

Khari, M. (2016). *Embedding security in Software Development Life Cycle (SDLC) .* IEEE Xplore Document. Obtenido de http://ieeexplore.ieee.org/document/7724651

Laínez Fuentes, J. R., Noriega Martínez, R., Ramos, D., & Durango, A. (2015). *Curso de Ingeniería de Software.* IT Campus Academy.

Loic, F. A. (2005). *Introducción a la Ingeniería del software: Modelos de Desarrollo de Programas.* Delta.

M., J. V. (2010). *Preservación documental digital y seguridad informática.* México: Investigación bibliotecológica.

Macconnel, S. (1996). *Desarrollo y Gestión de proyectos Informáticos.* Madrid: Macgraw-Hill.

Milano, P. (2007). *Seguridad en el ciclo de vida del desarrollo de software*. Obtenido de http://www.cybsec.com/upload/cybsec_Tendencias2007_Seguridad_SDLC.pdf

Mohammadi, M., Chu, B., Lipford, H. R., & Murphy, H. (2016). Automatic web security unit testing. *1th International Workshop on Automation of Software Test.* Obtenido de https://doi.org/10.1145/2896921.2896929

Nguyen, T. K., & Hwang, S. O. (2017). Large-Scale Detection of DOM-Based XSS Based on Publisher and Subscriber Model. Proceedings. *2016 International Conference on Computational Science and Computational Intelligence.* Obtenido de https://doi.org/10.1109/CSCI.2016.0187

Owasp.org. (2017). *OWASP top 10 - 2017.* Owasp.Org. Obtenido de https://doi.org/10.1002/kin.550040606

Pressman, S. R. (2010). *Ingeniería del Software: Un enfoque práctico* (7° ed.). México: The McGraw-Hill Companies.

Rakitin, S. R. (2001). *Software Verification and Validation for Practitioners and Managers.* Estados Unidos: Artech House Print on Demand.

Ramos, D. (2015). *Curso de Ingeniería de Software.* Copyright.

Ruse, M., & Basu, S. (2013). Detecting cross-site scripting vulnerability using concolic testing. *10th International Conference on Information Technology: New Generations.* Obtenido de https://doi.org/10.1109/ITNG.2013.97

Sánchez, S., Sicilia, M. Á., & Rodríguez, D. (2012). *Ingeniería del Software un enfoque desde la guía SWEBOK.* Madrid: Alfaomega.

Schwaber, K., & Sutherland, J. (2011). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game.* Croacia: Scrum.org.

Shehab Farhan, A. R., & Mostafa Mostafa, G. M. (2018). *A Methodology for Enhancing Software Security during Development Processes.* 21st Saudi Computer Society National Computer Conference. Obtenido de https://doi.org/10.1109/NCG.2018.8593135

Shuaibu, M. B., & Ibrahim, R. A. (2018). Web application development model with security concern in the entire life-cycle. *4th IEEE International Conference on Engineering Technologies and Applied Sciences.* Obtenido de https://doi.org/10.1109/ICETAS.2017.8277849

Sommerville, I. (2005). *Ingeniería del Software* (7° ed.). Madrid: PEARSON EDUCACIÓN. S.A.

Turner, R., & Jain, A. (2002). *Agile Meets CMMI: Culture Clash or Common Cause?* XP/Agile Universe.

ANEXOS

Anexo 1 Resolución de aprobación del proyecto de investigación

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N° 0933-2019/FIAU-USS

Chiclayo, 22 de julio de 2019

VISTO:

El Acta de Reunión N° de fecha 22 de julio de 2019,, para la ejecución de la Tesis titulada: *"MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MAS RECURRENTES"*, presentada por el(los) estudiante(s) **DIAZ CASTAÑEDA RUDY SLAYTONW** de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS** y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: *"La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";*

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el Proyecto de Tesis denominado ***"MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERISTICAS DE SEGURIDAD PARA VULNERABILIDADES MAS RECURRENTES"***, perteneciente a la Línea de Investigación **INGENIERÍA DE SOFTWARE - PROCESO DE DESARROLLO DE SOFTWARE,** a cargo del(los) estudiante(s) **DIAZ CASTAÑEDA RUDY SLAYTONW,** de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS.**

ARTÍCULO 2°: ESTABLECER, que la inscripción de la Tesis se realice a partir de emitida la presente resolución, y tendrá una vigencia máxima de 02 años.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE

Cc: Dirección de Investigación, CPGYT, Interesados, Archivo

Anexo 2 Perfiles profesionales de los evaluadores – método Delphi

EVALUADOR 01

RONALD RAMIREZ BLANCO

Experto en Seguridad Informática en el Ministerio de Relaciones Exteriores - Perú

EXPERIENCIA LABORAL

ANALISTA DE SEGURIDAD INFORMÁTICA EN TOURING Y AUTOMOVIL CLUB DEL PERÚ, COORDINADOR DE SEGURIDAD INFORMÁTICA EN PCM PRESIDENCIA DEL CONSEJO DE MINISTROS DEL PERÚ, GESTOR DE PROYECTOS EN VILSOL, PROFESIONAL SGSI EN CONSORCIO GMD S.A., ADMINISTRADOR DE SEGURIDAD EN GMD, AUDITOR Y CONSULTOR DE TI EN ACYSTEMS CONSULTING, CONSULTOR DE CIBERSEGURIDAD Y SEGURIDAD DE LA INFORMACIÓN EN BARCO PERU, ADMINSITRADOR DE SEGURIDAD INFORMÁTICA EN GMD, ESPECIALISTA EN SEGURIDAD INFORMÁTICA EN OEFA,
ESPECIALISTA EN SEGURIDAD DE LA INFORMACIÓN EN PODER JUDICIAL DEL PERÚ.

CERTIFICACIONES

CERTIFIED NETWORK SECURITY SPECIALIST, ISO 27001, LEAD AUDITOR (I27001LA), COBIT 5.0 FOUNDATION, ISO 22301 AUDITOR, RISK MANAGER ISO 31000, AUDITOR INTERNO ISO 27001:2013, PRIVACY

	AND DATA PROTECTION, CERTIFIED ETHICAL HACKER – C|EH

EVALUADOR 02

	**GINO BEGAZO** Analista Jr. de Seguridad de la Información en Centria - CSC Grupo Breca Egresado de la Universidad de Lima **EXPERIENCIA LABORAL** ANALISTA JR SEGURIDAD DE INFORMACIÓN BBVA, ANALISTA JR DE SEGUIRIDAD DE LA INFORMACIÓN EN CENTRIA – CSC GRUPO BRECA **CERTIFICACIONES** ITIL 4 FOUNDATION CERTIFICATE, EDX HONOR CODE CERTIFICATE LINUX

EVALUADOR 03

<table>
<tr>
<td>  </td>
<td>CHRISTIAN TAIPE SEDANO
Líder de seguridad informática en Cencosud S.A.
Egresado de la UTP Universidad Tecnológica del Perú.

EXPERIENCIA LABORAL
OPERADOR DE SERVICIO DE SEGURIDAD INFORMÁTICA EN BAFING, ESPECIALISTA EN SEGURIDAD INFORMÁTICA EN BAFING, ESPECIALISTA SENIOR EN SEGURIDAD INFORMÁTICA EN BAFING. LÍDER DE SEGURIDAD INFORMÁTICA EN CENCOSUD S.A.

CERTIFICACIONES
SCRUM FOUNDATIONS PROFESSIONAL CERTIFICATE, CERTIFIED ETHICAL HACKER, MCAFEE CERTFIED PRODUCT SPECIALIST</td>
</tr>
</table>

Anexo 3 Validación de instrumento por Juicio de Expertos

VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS GENERALES

1.1. **Apellidos y Nombres del experto**: Fernandez Bances Dennis Dario

1.2. **Cargo e Institución donde labora:** Gerente General – Empresa Unimatik

1.3. **Nombre del Instrumento a evaluar:** Cuestionario

1.4. Autor del instrumento: Rudy Slaytonw Díaz Castañeda

1.5. Título de la investigación: "MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES"

ÍTEM	CRITERIOS A EVALUAR										Observaciones (si debe eliminarse o modificarse un ítem por favor indique)
	Claridad en la redacción		Coherencia interna		Inducción a la respuesta (Sesgo)		Lenguaje adecuado con el nivel de informante		Mide lo que pretende		
	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	
1.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?	x		x		x		x		x		

2.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de implementación?	x		x		x		x		x		
3.- Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?	x		x		x		x		x		

4.- Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?	x		x		x		x		x		
5.- Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?	x		x		x		x		x		

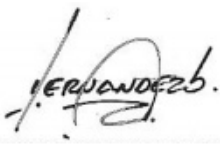
6.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?	x		x		x		x		x		
7.- Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?	x		x		x		x		x		
8.- Según el modelo de procesos propuesto. ¿Considera usted	x		x		x		x		x		

que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?											
9.- Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?	x		x		x		x		x		
10.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los	x		x		x		x		x		

desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?											

ASPECTOS GENERALES	**SÍ**	**NO**	……
El Instrumento contiene instrucciones claras y precisas para responder el cuestionario.	x		
Los ítems permiten el logro del objetivo de la investigación.	x		
Los ítems están distribuidos en forma lógica y secuencial.	x		
El número de ítems es suficiente para recoger la información. En caso de ser negativa su respuesta, sugiera los ítems a añadir.	x		

VALIDEZ			
APLICABLE	x	NO APLICABLE	
APLICABLE ATENDIENDO LAS OBSERVACIONES			

Firma: 	**DNI:** 43110190	**Fecha:** 10/07/2020
	Teléfono: 945028551	**E-mail:** dennis.fernandez.bances@gmail.com

VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS GENERALES

1.1. Apellidos y Nombres del experto: Arcila Diaz Juan Carlos

1.2. Cargo e Institución donde labora: Cognity S.A.C.

1.3. Nombre del Instrumento a evaluar: Cuestionario

1.4. Autor del instrumento: Rudy Slaytonw Díaz Castañeda

1.5. Título de la investigación: "MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES"

ÍTEM	CRITERIOS A EVALUAR										Observaciones (si debe eliminarse o modificarse un ítem por favor indique)
	Claridad en la redacción		Coherencia interna		Inducción a la respuesta (Sesgo)		Lenguaje adecuado con el nivel de informante		Mide lo que pretende		
	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	
1.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?	x		x		x		x		x		
2.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo	x		x		x		x		x		

coste de implementación?											
3.- Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?	x		x		x		x		x		
4.- Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para	x		x		x		x		x		

reducir las vulnerabilidades de seguridad más recurrentes?											
5.- Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?	x		x		x		x		x		
6.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?	x		x		x		x		x		
7.- Según el modelo de procesos propuesto. ¿Según su	x		x		x		x		x		

experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?											
8.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?	x		x		x		x		x		
9.- Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?	x		x		x		x		x		

10.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?	x		x		x		x		x	

ASPECTOS GENERALES	SÍ	NO	.....
El Instrumento contiene instrucciones claras y precisas para responder el cuestionario.	x		
Los ítems permiten el logro del objetivo de la investigación.	x		
Los ítems están distribuidos en forma lógica y secuencial.	x		
El número de ítems es suficiente para recoger la información. En caso de ser negativa su respuesta, sugiera los ítems a añadir.	x		

VALIDEZ			
APLICABLE	x	NO APLICABLE	
APLICABLE ATENDIENDO LAS OBSERVACIONES			

Firma: JUAN CARLOS ARCILA DIAZ INGENIERÓ DE SISTEMAS REG. CIP. 187605	**DNI:** 47715777	**Fecha:** 10/07/2020
	Teléfono: 931742904	**E-mail:** incanatoapps@gmail.com

VALIDACIÓN DE INSTRUMENTOS POR JUICIO DE EXPERTOS

I. DATOS GENERALES

1.1. Apellidos y Nombres del experto: Dávila Fuentes Juan Carlos

1.2. Cargo e Institución donde labora: BIGDAVI S.A.C.

1.3. Nombre del Instrumento a evaluar: Cuestionario

1.4. Autor del instrumento: Rudy Slaytonw Díaz Castañeda

1.5. Título de la investigación: "MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES"

ÍTEM	CRITERIOS A EVALUAR										Observaciones (si debe eliminarse o modificarse un ítem por favor indique)
	Claridad en la redacción		Coherencia interna		Inducción a la respuesta (Sesgo)		Lenguaje adecuado con el nivel de informante		Mide lo que pretende		
	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	SÍ	NO	
1.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?	x		x		x		x		x		
2.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de	x		x		x		x		x		

implementación?											
3.- Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?	x		x		x		x		x		
4.- Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad	x		x		x		x		x		

más recurrentes ?											
5.- Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?	x		x		x		x		x		
6.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?	x		x		x		x		x		
7.- Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad	x		x		x		x		x		

informática en cuál fase se debe darle más importancia a la seguridad?											
8.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?	x		x		x		x		x		
9.- Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?	x		x		x		x		x		
10.- Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los	x		x		x		x		x		

desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?											

ASPECTOS GENERALES	SÍ	NO	.....
El Instrumento contiene instrucciones claras y precisas para responder el cuestionario.	x		
Los ítems permiten el logro del objetivo de la investigación.	x		
Los ítems están distribuidos en forma lógica y secuencial.	x		
El número de ítems es suficiente para recoger la información. En caso de ser negativa su respuesta, sugiera los ítems a añadir.	x		

VALIDEZ			
APLICABLE	x	NO APLICABLE	
APLICABLE ATENDIENDO LAS OBSERVACIONES			

Firma: 	**DNI:** 42928629	**Fecha:** 10/07/2020
	Teléfono: 941172727	**E-mail:** jucadafu@gmail.com

Anexo 4 Cuestionario de preguntas para medir la viabilidad de la propuesta.

CUESTIONARIO PARA MEDIR LA VIABILIDAD DEL MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador para que la aplicación del método Delphi sea correcto.
- Complete con una “X” en alguna de las dos alternativas “A” o “B”, luego se le solicita redactar su respuesta a la pregunta “¿Por qué?” en las casillas de color celeste.

PREGUNTA 1

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 2

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de implementación?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 3

Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 4

Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 5

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?

A	**SI**	

B	**NO**	

¿POR QUÉ?

PREGUNTA 6

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 7

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?

Fase de Requerimientos	
Fase de Análisis y Diseño	
Fase de Implementación y Codificación	
Fase de Pruebas	
Fase de Lanzamiento	

¿POR QUÉ?

PREGUNTA 8

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 9

Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?

A	**SI**	
B	**NO**	

¿POR QUÉ?

PREGUNTA 10

Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?

A	**SI**	
B	**NO**	

¿POR QUÉ?

Aquí culminan sus primeras respuestas a este cuestionario.

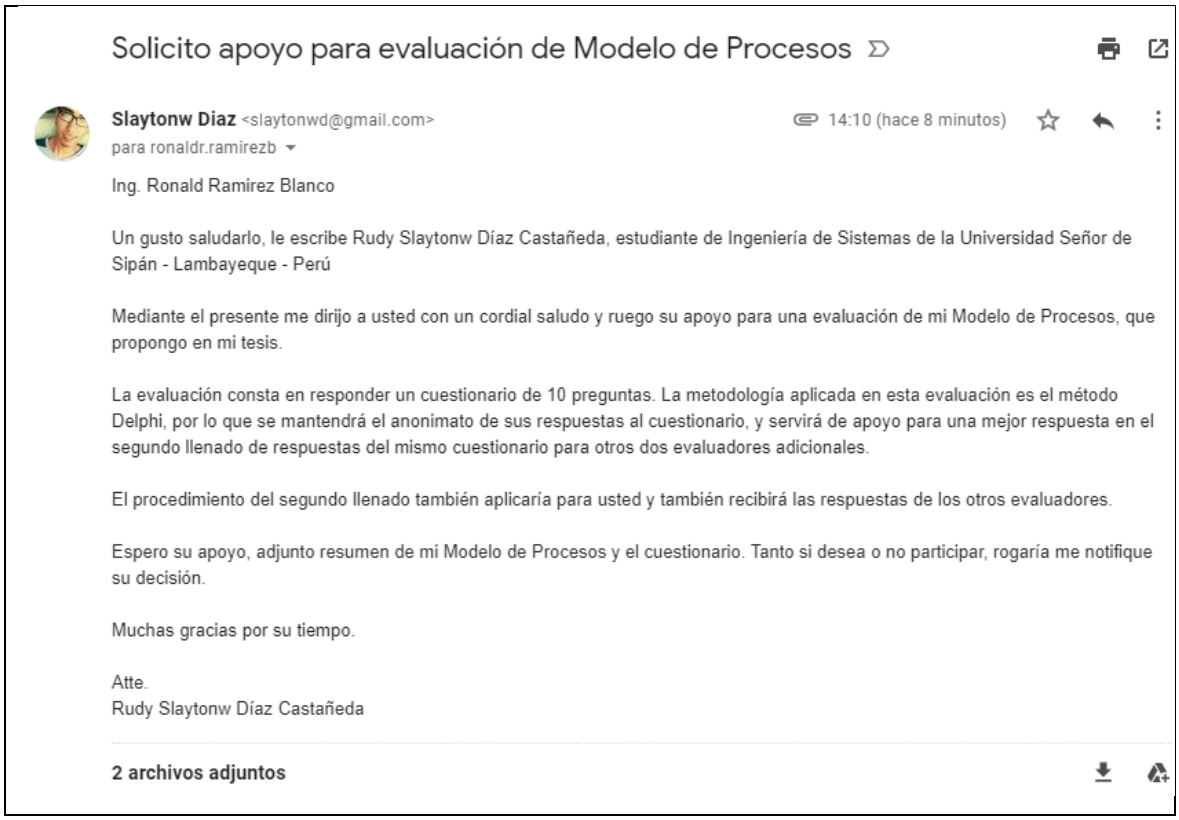
Siguiendo el método Delphi, se le volverá a enviar este documento con sus respuestas, además con las respuestas de otros evaluadores para que sirvan de retroalimentación o reflexión a sus primeras respuestas, y así nuevamente sus respuestas sean mejoradas o reafirmadas. Esto permitirá una afinación y consenso de respuestas por parte de todos los evaluadores.

Muchas gracias por su ayuda.

Rudy Slaytonw Díaz Castañeda.

Anexo 5 Comunicación con los expertos de evaluación.

EVALUADOR 01



Solicito apoyo para evaluación de Modelo de Procesos

Slaytonw Diaz <slaytonwd@gmail.com> 14:10 (hace 8 minutos)
para ronaldr.ramirezb

Ing. Ronald Ramirez Blanco

Un gusto saludarlo, le escribe Rudy Slaytonw Díaz Castañeda, estudiante de Ingeniería de Sistemas de la Universidad Señor de Sipán - Lambayeque - Perú

Mediante el presente me dirijo a usted con un cordial saludo y ruego su apoyo para una evaluación de mi Modelo de Procesos, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que se mantendrá el anonimato de sus respuestas al cuestionario, y servirá de apoyo para una mejor respuesta en el segundo llenado de respuestas del mismo cuestionario para otros dos evaluadores adicionales.

El procedimiento del segundo llenado también aplicaría para usted y también recibirá las respuestas de los otros evaluadores.

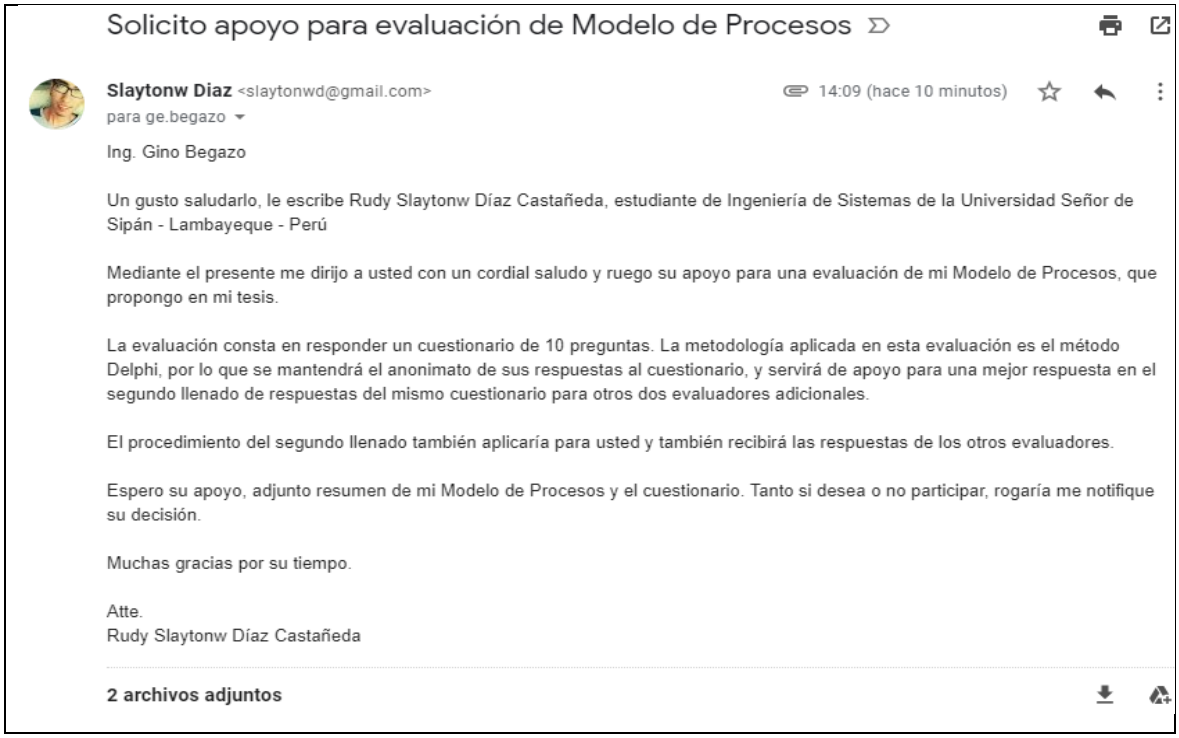
Espero su apoyo, adjunto resumen de mi Modelo de Procesos y el cuestionario. Tanto si desea o no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte.
Rudy Slaytonw Díaz Castañeda

2 archivos adjuntos

EVALUADOR 02



Solicito apoyo para evaluación de Modelo de Procesos

Slaytonw Diaz <slaytonwd@gmail.com> 14:09 (hace 10 minutos)
para ge.begazo

Ing. Gino Begazo

Un gusto saludarlo, le escribe Rudy Slaytonw Díaz Castañeda, estudiante de Ingeniería de Sistemas de la Universidad Señor de Sipán - Lambayeque - Perú

Mediante el presente me dirijo a usted con un cordial saludo y ruego su apoyo para una evaluación de mi Modelo de Procesos, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que se mantendrá el anonimato de sus respuestas al cuestionario, y servirá de apoyo para una mejor respuesta en el segundo llenado de respuestas del mismo cuestionario para otros dos evaluadores adicionales.

El procedimiento del segundo llenado también aplicaría para usted y también recibirá las respuestas de los otros evaluadores.

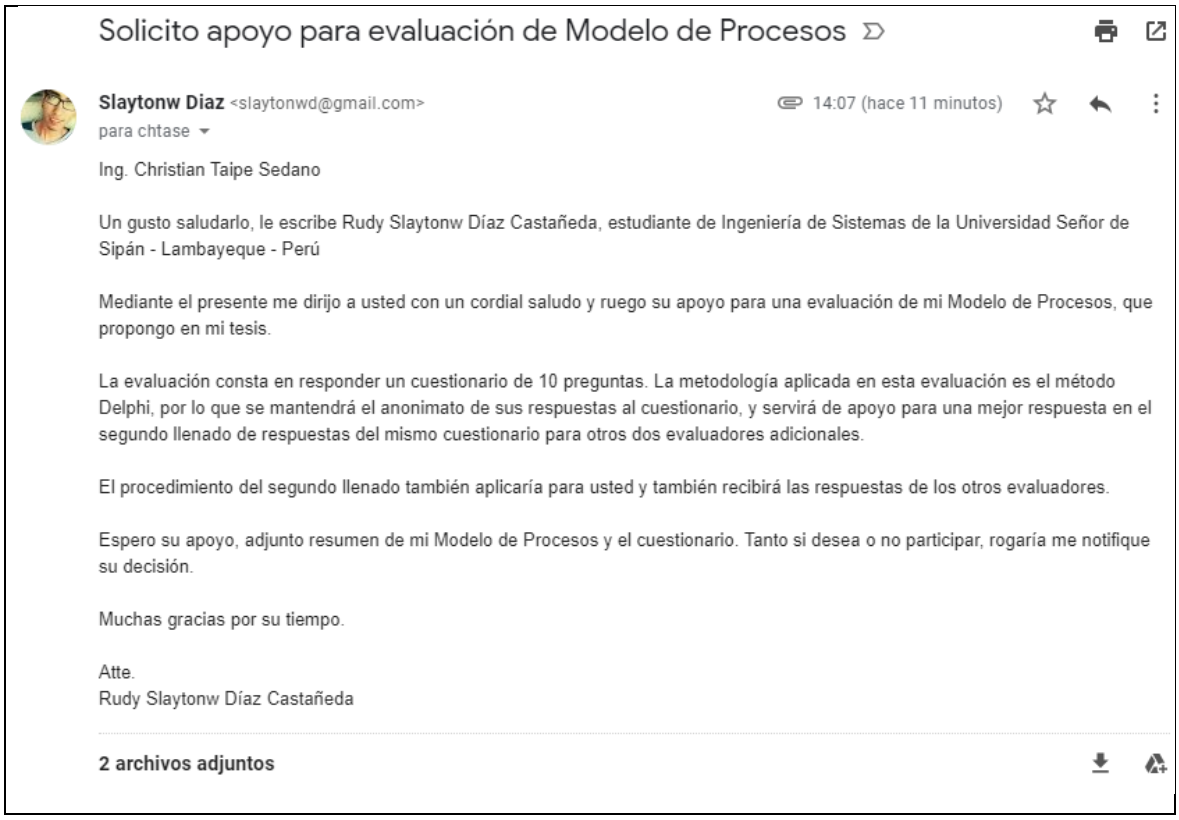
Espero su apoyo, adjunto resumen de mi Modelo de Procesos y el cuestionario. Tanto si desea o no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte.
Rudy Slaytonw Díaz Castañeda

2 archivos adjuntos

EVALUADOR 03



Solicito apoyo para evaluación de Modelo de Procesos

Slaytonw Diaz <slaytonwd@gmail.com> 14:07 (hace 11 minutos)
para chtase

Ing. Christian Taipe Sedano

Un gusto saludarlo, le escribe Rudy Slaytonw Díaz Castañeda, estudiante de Ingeniería de Sistemas de la Universidad Señor de Sipán - Lambayeque - Perú

Mediante el presente me dirijo a usted con un cordial saludo y ruego su apoyo para una evaluación de mi Modelo de Procesos, que propongo en mi tesis.

La evaluación consta en responder un cuestionario de 10 preguntas. La metodología aplicada en esta evaluación es el método Delphi, por lo que se mantendrá el anonimato de sus respuestas al cuestionario, y servirá de apoyo para una mejor respuesta en el segundo llenado de respuestas del mismo cuestionario para otros dos evaluadores adicionales.

El procedimiento del segundo llenado también aplicaría para usted y también recibirá las respuestas de los otros evaluadores.

Espero su apoyo, adjunto resumen de mi Modelo de Procesos y el cuestionario. Tanto si desea o no participar, rogaría me notifique su decisión.

Muchas gracias por su tiempo.

Atte.
Rudy Slaytonw Díaz Castañeda

2 archivos adjuntos

Anexo 6 Evidencia de cuestionario aplicado al evaluador 1

CUESTIONARIO PARA MEDIR LA VIABILIDAD DEL MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador para que la aplicación del método Delphi sea correcto.
- Complete con una “X” en alguna de las dos alternativas “A” o “B”, luego se le solicita redactar su respuesta a la pregunta “¿Por qué?” en las casillas de color celeste.

PREGUNTA 1

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Tiene las Fases básicas y esenciales que todo aplicativo debe contar para un proceso de construcción de Software

PREGUNTA 2

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de implementación?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Existen herramientas y procedimientos de libre acceso, los cuales coadyuvan a cualquier sistema a implementarse.

PREGUNTA 3

Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

No se tiene especificado el análisis de Costo-Beneficio y tiempos

PREGUNTA 4

Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

Solo cumple lo básico y necesario, no se están considerando puntos como entrenamiento en seguridad y pruebas de Penetracion Testing.

PREGUNTA 5

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Considero que se puede mejorar en base a lo que se ha propuesto y elaborado, se puede desplegar y añadir actividades.

PREGUNTA 6

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Contiene las pautas básicas y escenciales del SDLC-S, pero considero que se debería profundizar mas en las etapas de implementación y lanzamiento.

PREGUNTA 7

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?

Fase de Requerimientos	**X**
Fase de Análisis y Diseño	**X**
Fase de Implementación y Codificación	
Fase de Pruebas	
Fase de Lanzamiento	

¿POR QUÉ?

Todas las Fases son importantes en su magnitud, pero sin embargo, las mejores prácticas recomiendan que las 1eras fases son importantes para posteriormente no acarrear costos de diseño, mano de obra y redefinición de tiempos, es por ello, que las fases de requerimiento y análisis y diseño son las que deben ser mejor trabajadas.

PREGUNTA 8

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Contiene las Fases primarías, solo habría considerar la ultima etapa , donde se incluiría el mantenimiento de SW

PREGUNTA 9

Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

Se tendría que entrar un poco más a detalle para otorgar una opinión a los productos resultantes.

PREGUNTA 10

Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

El Modelo Propuesto contiene actividades básicas y esenciales sobre las características de seguridad, pero podría profundizarse en las fases de Implementación y lanzamiento, sin embargo no contiene una metodología ágil de desarrollo y considero que esto podría mejorar

Aquí culminan sus primeras respuestas a este cuestionario.
Siguiendo el método Delphi, se le volverá a enviar este documento con sus respuestas, además con las respuestas de otros evaluadores para que sirvan de retroalimentación o reflexión a sus primeras respuestas, y así nuevamente sus respuestas sean mejoradas o reafirmadas. Esto permitirá una afinación y consenso de respuestas por parte de todos los evaluadores.
Muchas gracias por su ayuda.
Rudy Slaytonw Díaz Castañeda.

Anexo 7 Evidencia de aplicación de cuestionario al evaluador 2

CUESTIONARIO PARA MEDIR LA VIABILIDAD DEL MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador para que la aplicación del método Delphi sea correcto.
- Complete con una “X” en alguna de las dos alternativas “A” o “B”, luego se le solicita redactar su respuesta a la pregunta “¿Por qué?” en las casillas de color celeste.

PREGUNTA 1

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Considero que sí ya que el modelo no es complejo de entender, no creo que dependa del tamaño del equipo de desarrollo ya que las actividades lo pueden realizar un programador promedio como avanzado y las actividades pueden ser desarrolladas por uno o varios integrantes de un equipo de desarrollo. Muchas veces no se implementa ya que no hay un modelo o una estructura de trabajo a seguir, contando con un modelo que no implica mayor complejidad sólo aumentar más actividades con respecto a la seguridad que beneficiará a corto o largo plazo el éxito de un proyecto.

PREGUNTA 2

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de implementación?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

No considero de bajo costo, ya que eso dependerá del alcance del proyecto a realizar considero que de aplicar este modelo aumentaría del 10% al 30% el tiempo de desarrollo y por tal un aumento en presupuesto. Pero el beneficio se verá reflejado luego de la publicación o en etapa de producción reduciendo actividades de testeo o inconvenientes que puedan surgir en cuanto a seguridad, reduciendo costos post producción en actividades de mantenimiento correctivas.

PREGUNTA 3

Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Considero que sí ya que al usar estas herramientas se reduce tiempo en actividades de desarrollo o mantenimiento del software. Y económicamente ya que reducirá en el futuro presupuesto en mantenimientos correctivos y de forma indirecta reducción de daños causados por temas de seguridad.

PREGUNTA 4

Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Considero que sí desde el punto de vista general, pero eso dependerá del proyecto de desarrollo en cuestión teniendo en cuenta el tipo de información que se guarda si es muy importante o sensible, la cantidad de usuarios que acceden al software, si es interna o pública, si es más propensa a ser atacada, etc. Según el nivel de seguridad considero que se agreguen más actividades, pero no reducir.

PREGUNTA 5

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?

A	**SI**	
B	**NO**	**X**

¿POR QUÉ?

Pienso que son los básicos e importantes que se deberían implementar.

PREGUNTA 6

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?

A	**SI**	**X**

B	**NO**	

¿POR QUÉ?

Considero que sí, la mayoría de proyectos de desarrollo solo lo aplica en la etapa de requerimientos, pero no en la etapa de análisis y diseño o en la implementación.

PREGUNTA 7

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?

Fase de Requerimientos	
Fase de Análisis y Diseño	
Fase de Implementación y Codificación	
Fase de Pruebas	**X**
Fase de Lanzamiento	

¿POR QUÉ?

Pienso que en la fase de pruebas se debe dar más importancia, ya que es la fase crítica de todo proyecto de software.

PREGUNTA 8

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Pienso que sí la mayoría de metodologías de desarrollo de software manejan tres fases principales requerimientos, análisis, desarrollo, pruebas y lanzamiento, según la metodología en cualquier de las etapas hay actividades a desarrollar.

PREGUNTA 9

Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Si, ya que los productos entregados serán como un mapa para el equipo de desarrollo de las acciones realizadas en cada módulo con respecto a la seguridad del software.

PREGUNTA 10
Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?

A	**SI**	**X**
B	**NO**	

¿POR QUÉ?

Considero que sí, ya que si se realizan todas las actividades se aseguran tener un software que reduzca los inconvenientes con respecto a seguridad, que muchas veces se deja de lado, pero implementando este modelo también se reducirá a corto o largo plazo actividades correctivas.

Aquí culminan sus primeras respuestas a este cuestionario.
Siguiendo el método Delphi, se le volverá a enviar este documento con sus respuestas, además con las respuestas de otros evaluadores para que sirvan de retroalimentación o reflexión a sus primeras respuestas, y así nuevamente sus respuestas sean mejoradas o reafirmadas. Esto permitirá una afinación y consenso de respuestas por parte de todos los evaluadores.
Muchas gracias por su ayuda.
Rudy Slaytonw Díaz Castañeda.

Anexo 8 Evidencia de aplicación de cuestionario al evaluador 3

CUESTIONARIO PARA MEDIR LA VIABILIDAD DEL MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES

Consideraciones:

- Las respuestas se mantendrán en el anonimato bajo el conocimiento del investigador para que la aplicación del método Delphi sea correcto.
- Complete con una “X” en alguna de las dos alternativas “A” o “B”, luego se le solicita redactar su respuesta a la pregunta “¿Por qué?” en las casillas de color celeste.

PREGUNTA 1

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es viable para pequeñas empresas?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

A pesar de tener fases de implementación que pueden ser bloqueantes y tardar más tiempo de lo previsto, teniendo un flujo exitoso se puede considerar viable

PREGUNTA 2

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto es de bajo coste de implementación?

A	**SI**	
B	**NO**	x

¿POR QUÉ?

Dentro del modelo no especifica si el software usado es libre, además al ser un modelo por capas consecutivas podría demandar tiempo de implementación.

PREGUNTA 3

Según el modelo de procesos propuesto. ¿Considera usted que en las fases de codificación y pruebas del modelo propuesto los análisis estáticos y dinámicos con software automatizado ayudan a las pequeñas empresas a reducir tiempo y dinero?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

Considerando que el modelo se aplica al final del desarrollo y se ha estimado un tiempo para corregir los errores, solo si los errores no son graves se puede considerarse que ayudan a reducir tiempo y dinero

PREGUNTA 4

Según el modelo de procesos propuesto. ¿Considera usted que los procesos definidos en el modelo propuesto son suficientes para reducir las vulnerabilidades de seguridad más recurrentes?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

Para una pequeña empresa con un software limitado si considero que sea un modelo suficiente para reducir las vulnerabilidades más recurrentes. Además, debe considerarse el entorno en el que se desplegara el software sea el caso interno o externo.

PREGUNTA 5

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática retiraría o agregaría algún proceso al modelo propuesto?

A	**SI**	
B	**NO**	x

¿POR QUÉ?

Dado el modelo y la planificación del mismo lo considero suficiente

PREGUNTA 6

Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto asegura características de seguridad en cada una de las etapas del desarrollo de software?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

Si ya que las fases del modelo están detalladas y definen su aplicación.

PREGUNTA 7

Según el modelo de procesos propuesto. ¿Según su experiencia en seguridad informática en cuál fase se debe darle más importancia a la seguridad?

Fase de Requerimientos	
Fase de Análisis y Diseño	x
Fase de Implementación y Codificación	x
Fase de Pruebas	
Fase de Lanzamiento	

¿POR QUÉ?

Son las fases que considero clave para llevar a cabo un buen análisis y posterior revisión de bugs de seguridad

PREGUNTA 8
Según el modelo de procesos propuesto. ¿Considera usted que el modelo propuesto se adapta a la mayoría de metodologías de desarrollo de software en cada una de sus fases?

A	**SI**	
B	**NO**	x

¿POR QUÉ?

Debido a que las metodologías de desarrollo son muy diversas y el modelo planteado es por fases y aplicado al final de la implementación del producto, esto dificulta la adaptación del modelo a metodologías mas actuales.

PREGUNTA 9
Según el modelo de procesos propuesto. ¿Considera usted que los productos resultantes del modelo propuesto son de utilidad para el desarrollo del software?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

Solo si es aplicado a determinados productos según una metodología de desarrollo implementada en fases.

PREGUNTA 10
Según el modelo de procesos propuesto. ¿Considera usted que el modelo ayuda a los desarrolladores de manera correcta y ágil a desarrollar software con características de seguridad?

A	**SI**	x
B	**NO**	

¿POR QUÉ?

Solo si es aplicado a determinados productos desarrollado bajo metodologías realizadas por fases.

Aquí culminan sus primeras respuestas a este cuestionario.
Siguiendo el método Delphi, se le volverá a enviar este documento con sus respuestas, además con las respuestas de otros evaluadores para que sirvan de retroalimentación o reflexión a sus primeras respuestas, y así nuevamente sus respuestas sean mejoradas o reafirmadas. Esto permitirá una afinación y consenso de respuestas por parte de todos los evaluadores.
Muchas gracias por su ayuda.
Rudy Slaytonw Díaz Castañeda.