



UNIVERSIDAD TÉCNICA DE MACHALA

FACULTAD DE INGENIERÍA CIVIL  
CENTRO DE POSTGRADOS  
MAESTRÍA EN SOFTWARE

PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON  
ÉNFASIS EN LA SEGURIDAD

Ing. GALO YOVANY LÓPEZ AJILA

MACHALA

2021



UNIVERSIDAD TÉCNICA DE MACHALA

FACULTAD DE INGENIERÍA CIVIL  
CENTRO DE POSTGRADOS  
MAESTRÍA EN SOFTWARE

PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON  
ÉNFASIS EN LA SEGURIDAD

Ing. GALO YOVANY LÓPEZ AJILA

MACHALA

2021



**UNIVERSIDAD TÉCNICA DE MACHALA  
FACULTAD DE INGENIERÍA CIVIL  
CENTRO DE POSTGRADOS**

**MAESTRÍA EN SOFTWARE**

**PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON  
ÉNFASIS EN LA SEGURIDAD**

**Ing. GALO YOVANY LÓPEZ AJILA.**

**(PROPUESTA METODOLÓGICA Y TECNOLOGÍA AVANZADA EN OPCIÓN AL  
TÍTULO DE MAGISTER EN SOFTWARE)**

**TUTOR: ING. VÍCTOR LEWIS CHIMARRO CHIPANTIZA, MS**

**MACHALA**

**2021**

## DEDICATORIA

Dedico esta investigación, a Dios y a la Virgen del Cisne que han sido mis pilares fundamentales en momentos de desesperación y decaimiento, durante el tiempo que mantuve en este programa de maestría.

A mis queridos padres, Galo López Flores y Úrsula Ajila Ajila, a mis hermanos y sobrinos que han sido el apoyo incondicional, en cada momento de mi vida, siempre motivándome y confiando en mis capacidades.

A todas las personas que de alguna manera u otra me incentivaron a seguir en este proceso formación continua.

Galo Yovany López Ajila

## **AGRADECIMIENTO**

Un agradecimiento especial y sincero a todas las personas que de forma directa e indirecta ayudaron a la realización y desarrollo de este proyecto de investigación, a nuestros profesores que además de dictar sus cátedras en el aula supieron acogernos como sus amigos y darnos la guía necesaria para cumplir con nuestros objetivos trazados desde el primer día que pisamos las instalaciones universitarias; A nuestros Padres y hermanos, pilares fundamentales que siempre nos han brindado su aliento incondicional al momento de enfrentar este tipo de retos; al Master Lewis Chimarro y al Master Joffre Cartuche, tutor y cotutor respectivamente de esta Tesis que supieron dar las pautas que permitieron consolidar lo aprendido, a la master Jennifer Célleri, coordinadora de la maestría por la paciencia y entrega total en cada ayuda que se le solicito durante todo el transcurso de este programa académico en su I Cohorte, a todas las personas y directivos de esta universidad que nos supieron acoger como sus hijos y formarnos en el camino profesional.

Galo Yovany López Ajila

## **RESPONSABILIDAD DE AUTORÍA**

Por medio de la presente declaro ante el Comité Académico de la Maestría de Software, de la Universidad Técnica de Machala, que el trabajo de titulación, titulado “PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON ÉNFASIS EN LA SEGURIDAD”, de mi propia autoría, no contiene material escrito por otra persona al no ser referenciado debidamente en el texto; parte de ella o en su totalidad no ha sido aceptada para el otorgamiento de cualquier otro diploma de una institución nacional o extranjera.

Ing. Galo Yovany López Ajila

CI: 1103811913

Machala, 12 de marzo de 2021

## REPORTE DE SIMILITUD

---

### INFORME DE ORIGINALIDAD

---

**8%**

INDICE DE SIMILITUD

**7%**

FUENTES DE  
INTERNET

**0%**

PUBLICACIONES

**4%**

TRABAJOS DEL  
ESTUDIANTE

---

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

---

< 1%

★ Submitted to Universidad Técnica Nacional de Costa Rica

Trabajo del estudiante

---

---

Excluir citas

Activo

Excluir coincidencias

< 15 words

Excluir bibliografía

Activo

## **CERTIFICACION DEL TUTOR**

Por medio de la presente apruebo que el trabajo de titulación “PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON ÉNFASIS EN LA SEGURIDAD”, del autor Ing. Galo Yovany López Ajila, en opción al título que otorga la Master de Software, sea presentada al acto de defensa.

Ing. Víctor Lewis Chimarro Chipantiza Ms.

CI: 0703703413

Machala, 12 de marzo de 2021



## **CESIÓN DE DERECHOS DE AUTORIA**

Yo, GALO YOVANY LÓPEZ AJILA, en calidad de autor del presente trabajo titulado “PROPUESTA METODOLÓGICA DE DESARROLLO ÁGIL DE SOFTWARE CON ÉNFASIS EN LA SEGURIDAD”, autorizo a la UNIVERSIDAD TECNICA DE MACHALA, la publicación y distribución en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor declara que el contenido que se publicara es de carácter académico y enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Galo Yovany López Ajila

C.I. 1103811913

Machala, 12 de marzo de 2021

## **RESUMEN**

El mundo actual demanda de grandes productos de software, las empresas necesitan desarrollar con rapidez y tener entregables en el menor tiempo posible. Es ahí donde estas entidades introducen las metodologías de desarrollo ágil, en los proyectos que desarrollan, en la actualidad la seguridad se ha convertido en una tema muy importante dejando de ser un simple requerimiento y pasando a ser uno primordial dentro del ciclo de vida del mismo, la evolución de las amenazas y las vulnerabilidades va en aumento, por este motivo, es necesario la implementación de una metodología de desarrollo de software ágil que haga énfasis en la seguridad, no solo en etapas del desarrollo y despliegue, sino que se encuentre presente mediante técnicas, método y buenas prácticas de seguridad, en cada una de las fases del ciclo de vida del software. El objetivo principal, que tiene este trabajo de investigación, es el diseñar una metodología de desarrollo ágil de software que agregue el atributo de seguridad, mediante la aplicación de técnicas de seguridades informáticas, a fin de que nos permita minimizar vulnerabilidades frente a los ataques informáticos. Para esto lo primero que se hizo fue comparar diferentes metodologías de desarrollo ágil de software, luego diseñar la metodología de desarrollo ágil y por último validar la propuesta metodológica planteada; Los métodos y materiales utilizados para la obtención de la información fueron la Revisión Sistemática de la Literatura SLR y el estudio de caso. Para la obtención de la información, se realizó entrevista al equipo de desarrollo, ya que la población y muestra definida fue la empresa SOLNUS, de la ciudad de Loja, para la validación de esta información se utilizó la técnica de chi cuadrado la misma que permitió comprobar que la propuesta metodológica con énfasis en la seguridad permite minimizar las vulnerabilidades frente a ataques internos y externos. Obteniendo de esta forma un proyecto con resultados positivos que sirve de guía o base para futuros trabajos de investigación y que puede implementarse para proyectos de desarrollo

## **PALABRAS CLAVES**

Vulnerabilidades, metodologías, ingeniería de software, patrones de diseño, técnicas de seguridad, dimensión de la seguridad.

## **ABSTRACT**

Today's world demands great software products, companies need to develop quickly and have deliverables in the shortest possible time. It is there where these entities introduce agile development methodologies in the projects they develop, nowadays security has become a very important issue, ceasing to be a simple requirement and becoming a primary one within the life cycle, the evolution of threats and vulnerabilities is increasing, for this reason, it is necessary to implement an agile software development methodology that emphasizes security, not only in stages of development and deployment, but is present through techniques, method and good security practices, in each of the phases of the software life cycle. The main objective of this research work is to design an agile software development methodology that adds the security attribute, through the application of computer security techniques, in order to minimize vulnerabilities against computer attacks. The first step was to compare different agile software development methodologies, then to design the agile development methodology and finally to validate the proposed methodology; the methods and materials used to obtain the information were the Systematic Review of the SLR Literature and the case study. To obtain the information, an interview was conducted with the development team, since the population and sample defined was the SOLNUS company in the city of Loja. To validate this information, the chi-square technique was used, which allowed verifying that the methodological proposal with emphasis on security allows minimizing vulnerabilities to internal and external attacks. Thus, obtaining a project with positive results that serves as a guide or basis for future research work and can be implemented for development projects.

## **KEYWORDS**

Vulnerabilities, methodologies, software engineering, design patterns, security techniques, security dimensions

## ÍNDICE GENERAL

DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
RESPONSABILIDAD DE AUTORÍA .....	vi
REPORTE DE SIMILITUD.....	vii
CERTIFICACION DEL TUTOR.....	viii
CESIÓN DE DERECHOS DE AUTORIA.....	ix
RESUMEN .....	x
PALABRAS CLAVES.....	x
ABSTRACT .....	xi
KEYWORDS .....	xi
LISTA DE TABLAS Y FIGURAS .....	xvi
INTRODUCCIÓN.....	3
PLANTEAMIENTO DEL PROBLEMA .....	9
OBJETIVOS.....	10
Objetivo General: .....	10
Objetivos Específicos:.....	10
CAPÍTULO 1 MARCO TEÓRICO REFERENCIAL.....	11
1.1 ANTECEDENTES HISTÓRICOS DE LA INVESTIGACIÓN.....	11
1.1.1 Preguntas de Investigación .....	11
1.1.2 Proceso de Búsqueda .....	12
1.1.3 Criterios de Exclusión Y Inclusión.....	13
1.1.4 Grupos de Control .....	13
1.1.5 Cadena de Búsqueda.....	14
1.1.6 Selección de Estudios .....	15
1.1.7 Resultados de la Revisión.....	15
1.2 ANTECEDENTES CONCEPTUALES .....	18
1.2.1 Hipótesis de la Investigación.....	18
1.2.2 Categorización de las Variables.....	18
1.2.2.1 Variable Dependiente: .....	18
1.2.2.2 Variable Independiente:.....	18
1.2.3 Red de Categorías .....	18
1.2.4 Fundamentación Teórica de la Variable Independiente.....	19

1.2.4.1	Metodologías .....	19
1.2.4.2	Metodologías de Desarrollo Ágil.....	20
1.2.4.3	Seguridad en las Metodologías .....	21
1.2.4.4	Técnicas de Seguridades Informáticas .....	23
1.2.5	Fundamentación teórica de la variable dependiente .....	24
1.2.5.1	Vulnerabilidades.....	24
1.2.5.2	Tipos de Ataques.....	25
1.2.5.3	Alternativas de Solución.....	26
1.3	ANTECEDENTES CONTEXTUALES DE LA INVESTIGACIÓN .....	27
1.3.1	Delimitación del Contexto de Estudio.....	27
1.3.2	Propuesta de Solución y Contribuciones .....	27
1.3.3	Organización del Documento.....	29
CAPÍTULO 2 MATERIALES Y MÉTODOS .....		30
2.1	Tipo de Estudio o Investigación Realizada .....	30
2.2	Paradigma o enfoque en el cual se realizó .....	31
2.3	Cálculo de la Población y Muestra .....	33
2.4	Métodos Teóricos con los materiales utilizados .....	35
2.5	Métodos Empíricos con los materiales utilizados.....	38
2.6	Técnicas Estadísticas para el Procesamiento de Datos Obtenidos .....	38
CAPÍTULO 3 RESULTADOS OBTENIDOS .....		39
3.1	Selección de las Metodologías Ágiles para esta Investigación .....	39
3.2	Ciclo de Vida Propuesto .....	39
3.2.1	Explicación de las Fases de la Propuesta Metodológica .....	41
3.2.1.1	Fase de Análisis de Requisitos.....	41
3.2.1.2	Fase de Diseño .....	42
3.2.1.3	Fase de Desarrollo y Pruebas .....	42
3.2.1.4	Fase de Despliegue.....	42
3.3	Propuesta Metodológica, Ciclo de Vida, Fases que la conforman y Las Técnicas de Seguridad .....	43
3.3.1	Organización de la Propuesta Metodológica .....	43
3.3.2	Procesos de Trazabilidad de la Propuesta Metodológica .....	45
3.3.2.1	Explicación Del Diagrama De Trazabilidad De La Propuesta Metodológica .....	46
3.3.3	Descripción de las Técnicas Implementadas.....	49

a)	Establecer requisitos de seguridad .....	49
b)	Evaluación de los riesgos de seguridad.....	50
c)	Patrones de diseño.....	51
d)	Técnica De Seguridad Modelado De Amenazas.....	52
e)	Herramientas de código estático .....	52
f)	Herramientas de ofuscamiento de código.....	52
g)	Pruebas de penetración .....	53
h)	Pruebas de carga.....	53
i)	Plan de despliegue .....	54
3.4	Implementación de la Propuesta Metodológica.....	54
3.5	Ejecución de la Implementación .....	56
3.5.1	Estudio de Caso Práctico de la Metodología Propuesta .....	57
3.5.2	Fase de Análisis de Requisitos de Seguridad.....	57
3.5.3.1	Implementación del patrón de diseño MVC.....	63
3.5.3.2	Técnica De Seguridad Modelado De Amenazas TAM .....	67
3.5.3	Fase de Desarrollo y Pruebas .....	71
3.5.4	Fase de Despliegue.....	85
CAPÍTULO 4 DISCUSIÓN DE LOS RESULTADOS OBTENIDOS EN EL ESTUDUIO REALIZADO Y SU CORROBORACIÓN .....		87
4.1	Procesamiento de Datos y Corroboración de Resultados .....	87
4.2	Procesamiento Y Análisis para Comprobar la Necesidad y Aceptación de la Propuesta Metodológica con Énfasis en la Seguridad .....	89
4.3	Procesamiento y Análisis para Comprobar la Aceptación de las Técnicas, Métodos y Buenas Prácticas de Seguridad de la Propuesta Metodológica con Énfasis en la Seguridad .....	90
4.4	Prueba de Hipótesis Chi Cuadrado.....	92
4.4.1	Proceso de Obtención de las Frecuencias Esperadas y Observadas.....	93
4.4.1.1	Frecuencia Esperada .....	93
4.4.1.2	Cálculos Estadísticos del Chi Cuadrado Observado.....	95
4.4.1.3	Valor Calculado con la Técnica de chi cuadrado .....	95
4.4.1.4	Comparativa entre los Valores Observados Y Críticos.....	96
CONCLUSIONES.....		98
RECOMENDACIONES.....		99
BIBLIOGRAFIA.....		100
ANEXOS.....		104

Anexo 1 .....	104
Anexo 2 .....	106
Anexo 3 .....	108
Anexo 4 .....	110
Entrevista 1 .....	110
Anexo 5 .....	117
Entrevista 2 .....	117

## LISTA DE TABLAS Y FIGURAS

Tabla 1. Proceso de búsqueda, elaboración propia del autor .....	12
Tabla 2. Cadena de búsqueda y sus resultados, elaboración propia del autor .....	14
Tabla 3. Artículos que cumplen los criterios de inclusión, elaboración propia del autor .....	15
Tabla 4. Variables Causa - Efecto, elaboración propia del autor .....	32
Tabla 5. Tabla de metodologías y sus CDVS, Elaboración propia del autor .....	40
Tabla 6. Introducción de las técnicas de seguridad, elaboración propia del autor .....	44
Tabla 7. Técnicas de seguridad en las diferentes fases del CDVS, elaboración propia del autor ....	49
Tabla 8. Plantilla de Análisis de Requisitos de Seguridad, Elaboración propia del autor .....	49
Tabla 9. Plantilla de evaluación de riesgos de seguridad y privacidad, elaboración: propia del autor .....	50
Tabla 10. Roles y responsabilidades de la implementación de la propuesta metodológica, elaboración propia del autor .....	55
Tabla 11. Plan de Implementación, elaboración propia del autor.....	56
Tabla 12. Equipo de desarrollo, elaboración propia del autor .....	57
Tabla 13. Plantilla de Análisis de Requisitos de Seguridad .....	58
Tabla 14. Valoración de activos de la información, elaboración propia del autor .....	59
Tabla 15. Niveles de amenazas, elaboración propia del autor .....	59
Tabla 16. Niveles de Vulnerabilidades, elaboración propia del autor .....	60
Tabla 17. Niveles de Riesgos, elaboración propia del autor .....	60
Tabla 18. Tabla General de Evaluación de Riesgos, elaboración propia del autor .....	61
Tabla 19. Tabla General de Evaluación de Riesgos, elaboración propia del autor .....	68
Tabla 20. Análisis DREAD, elaboración propia del autor.....	70
Tabla 21. Explicación y Solución a la Vulnerabilidad, elaboración propia del autor.....	79
Tabla 22. Explicación y Solución a la Vulnerabilidad SQL INJECTION, elaboración propia .....	80
Tabla 23. Explicación y Solución a la Vulnerabilidad XSS REFLECTED, elaboración propia del autor	81
Tabla 24. Tabla de características del servidor, elaboración propia del autor .....	82
Tabla 25. Tabla de Transacciones exitosas y fallidas, elaboración propia del autor .....	84
Tabla 26. Valoración de preguntas, elaboración propia del autor .....	87
Tabla 27. Rangos de valoración de Cumplimiento y no Cumplimiento, elaboración propia del autor .....	88
Tabla 28. Resultados previa a la aplicación de la Propuesta Metodológica (Personal Técnico empresa SONUS, Loja) , elaboración propia del autor .....	89
Tabla 29. Resultados de la experiencia de la Propuesta Metodológica (Personal Técnico empresa SONUS, Loja) , elaboración propia del autor .....	91
Tabla 30. Propuesta Metodológica, Variable Independiente, elaboración propia del autor .....	93
Tabla 31. Variable Dependiente: Minimización de vulnerabilidades del software frente ataques informáticos, elaboración propia del autor .....	93
Tabla 32. Frecuencia Observada y Esperada según SPSS v25, elaboración propia del autor .....	94
Tabla 33. Pruebas de Chi Cuadrado, elaboración propia del autor .....	95



Figura 1. Red de categorías de las variables de investigación, elaboración propia del autor .....	19
Figura 2. Propuesta de la dimensión de la seguridad, elaboración propia.....	23
Figura 3. Ciclo de vida propuesta metodológica LG-SEG .....	28
Figura 4. Alcance y Tipo de la Investigación, elaboración propia .....	30
Figura 5. Enfoque Cuantitativo y los pasos que contiene, elaboración propia.....	31
Figura 6. Diseño de la Investigación – Tipos - Cuasi-Experimental – Características, Ventajas y Desventajas, elaboración propia.....	33
Figura 7. Organigrama funcional actual de la Empresa SOLNUS, ciudad de Loja, elaboración propia del autor.....	34
Figura 8. Esquema del tipo de Muestra, elaboración propia del autor .....	35
Figura 9. Técnica de recolección de datos, Entrevista, elaboración propia del autor.....	36
Figura 10. Ciclo de vida del Software de la Propuesto Metodológica, elaboración propia del autor .....	41
Figura 11. Proceso de Trazabilidad de la Propuesta Metodológica, elaboración propia del autor..	45
Figura 12. Patrón de Diseño MVC, elaboración propia del autor .....	51
Figura 13. Interfaz de la Herramienta WebGoat.....	53
Figura 14. Interfaz del Programa JMeter .....	54
Figura 15. Patrón de diseño Modelo – Vista – Controlador, elaboración propia del autor.....	62
Figura 16. Patrón de diseño implementado, elaboración propia del autor .....	63
Figura 17. Interfaz de usuario creada con el patrón MVC .....	64
Figura 18. Controlador creado con el patrón MVC.....	65
Figura 19. Clases creadas con el patrón MVC.....	66
Figura 20. Pantallas finales con el código creado por el patrón de diseño.....	66
Figura 21. Plantilla de Patrón de Diseño implementada .....	67
Figura 22. pantalla Principal de la Herramienta TAM, secciones de trabajo.....	69
Figura 23. Caso de Uso Autenticar usuario .....	69
Figura 24. Pantalla de cálculo de vulnerabilidades .....	70
Figura 25. Informe de como mitigar la vulnerabilidad.....	71
Figura 26. Pantalla de acceso del Programa Kiuwan .....	72
Figura 27. Pantalla de análisis de código del programa Kiuwan.....	72
Figura 28. Sumario de análisis del Programa Kiuwan .....	73
Figura 29. Pantalla principal de la herramienta Proguard .....	75
Figura 30. Código principal de ofuscamiento métodos, clases y campo .....	75
Figura 31. Interfaz del Programa Webgoat.....	76
Figura 32. Interfaz del Programa Burpsuite .....	77
Figura 33. Interfaz del Login del sistema .....	77
Figura 34. Ataque de Intercept .....	77
Figura 35. Cargas útiles que muestra el programa .....	78
Figura 36. Combinaciones de credenciales validas en el ataque .....	78
Figura 37. Página Principal del portal APN.EC.....	80
Figura 38. Ejecución del XSS Reflected .....	81
Figura 39. Ejecución del Programa JMeter .....	82
Figura 40. Ejecución de atributos.....	83
Figura 41. Imagen de las peticiones en JMeter.....	83

Figura 42. Imagen de línea de tiempo de la herramienta JMeter .....	84
Figura 43. Ciclo de vida propuesto para la etapa de despliegue, elaboración propia del autor .....	85
Figura 44. Representación Gráfica de los niveles de aceptación, previa a la aplicación de la Propuesta Metodológica (empresa SOLNUS, Loja), elaboración propia del autor.....	90
Figura 45. Representación Gráfica de los niveles de la experiencia de aceptación de las técnicas de la Metodología Propuesta, elaboración propia del autor.....	91
Figura 46. Grafico del Chi Cuadrado .....	96

## INTRODUCCIÓN

El mundo actual se encuentra adherido indudablemente a la tecnología, no obstante, el mismo uso continuo, concibe que existan riesgos en cada uno de los aplicativos que la misma tecnología posee. Es así que los riesgos tienden en aumentar a medida que evoluciona, como en la Web 2.0 [1].

En el transcurso de la última década con la creación de nuevas tecnologías, también se han creado nuevas amenazas, con lo que respecta el ciberespacio, los problemas principales que afectan con más celeridad el mundo actual son, los delitos informáticos, las operaciones producidas por la política y las intrusiones en redes comunicacionales. Estos tres elementos unidos ocasionan problemas serios en todos los países a nivel mundial, ya que actualmente todos los estados están interconectados entre sí, por tal motivo la inestabilidad de uno solo afecta el bienestar de todos los que lo rodean. La única forma de poder contrarrestar este problema sería la creación de una armonización conjunta de políticas de seguridad cibernética, entre todos los países afectados y así combatir los desafíos emergentes de la ciberseguridad, [2].

En América latina y el caribe, organizaciones mundiales como la OEA, promueven talleres enfocados en la creación de estas políticas, estos eventos están sentando las bases fundamentales para que en el futuro se pueda cambiar la conciencia sobre las amenazas que cada día van creciendo sobre todo el campo cibernético, estas bases auguran un futuro próspero y estable con el único fin de lograr una participación fuerte entre los países de la región.

Pero estos avances deben estar ligados estrictamente a leyes que penalicen estos actos, como se describe en líneas anteriores la armonización compartida permite aprender más sobre estas prácticas y así elevar la ciberseguridad en cada país de Sudamérica. Las leyes que se crean deben estar enfocadas de acuerdo a los convenios internacionales ofreciendo un marco legal integral, confiable y seguro para combatir estos delitos y no dejarlos desapercibidos. En la actualidad, un tercio de los países de Sudamérica y el caribe, han creado y fomentado leyes que permitan disminuir estos actos de ciberdelincuencia en áreas estratégicas como, la educación que han alcanzado niveles óptimos de madurez, introduciendo dentro de sus pensum de estudio una formación especializada en ciberseguridad, tanto en lo legal como en lo técnico. Los otros dos

tercios restantes muestran un escaso y casi nulo avance en estas áreas, en estos países es inexistente la seguridad digital solo se maneja de forma directa en el marco del campo técnico [2].

En el reporte de *Ciberseguridad, riesgos, avances y camino a seguir*, que ofrece el BID en su reporte del año 2020, se muestra estos avances en forma de dimensiones, las mismas se encuentran por niveles de madurez empezando por la *dimensión de las políticas y estrategias*, en la cual se plantean estrategias para consolidar la ciberseguridad nacional; la siguiente *dimensión es la de Cultura Cibernética y Sociedad*, en la cual se fomenta una mentalidad de confianza y mecanismos que permitan crear una conciencia responsable; *la dimensión de Capacitación y habilidades*, en la cual se desarrolla la sensibilización y la formación de profesionales en el área; *la dimensión Legal y Regulatoria*, creación de marcos legales y justicia penal y finalmente la *dimensión de Estándares y Tecnologías*, en la cual se mide la resiliencia, calidad y controles técnicos. Todas estas dimensiones unidas y enfocadas en un buen sentido proporcionan una guía práctica para poder crear conciencia social en cada país donde se las aplique.

En Ecuador, aún no se cuenta con estrategias de seguridad cibernética, se han obtenido avances significativos con la creación de entidades como *ECUCERT, ARCOTEL y el equipo de respuesta ante incidentes*. Estas entidades han permitido que el país avance en el adelanto para enfrentar las amenazas cibernéticas, en un estudio realizado en el 2018, el 50%, de las entidades privadas implementaron programas de concientización en ciberseguridades, en cambio el 70% de las empresas encuestadas, veían poco aceptable la efectividad de estos programas. En el marco legal los avances son mayores ya que la creación de leyes ha permitido que estos delitos sean penalizados. De acuerdo a este reporte Ecuador ha avanzado significativamente en las dimensiones de madurez establecidas en líneas anteriores, hasta el año 2020, ha alcanzado una media de nivel de grado tres, en todas las dimensiones [2].

Para Anchundia [3], en Ecuador, los ataques a la información confidencial están ganando notoriedad gracias al crecimiento del Internet. Los delitos cometidos son varios siendo los más comunes: los cambios de notas en instituciones educativas (universidades, colegios, institutos privados), hasta la falsificación de títulos de bachiller, tercer y cuarto nivel, así como, el robo y estafa de dinero de cuentas electrónicas, a todo esto, se lo conoce como ciberdelincuencia o delincuencia informática.

El Plan Nacional de Desarrollo Toda una Vida 2017 – 2021[4], propuesto por el gobierno nacional señala tres ejes de desarrollo: el eje 1, Derechos para todos durante toda la vida, el eje 2 Economía al servicio de la comunidad y el eje 3 Mas sociedad, mejor estado. El presente estudio se enmarca en el eje 2, específicamente en el objetivo 5: que promueve la productividad y competitividad para el crecimiento económico sostenible. Este objetivo a su vez enmarca varias políticas para este trabajo investigativo nos ubicamos en la política 5.6 que describe “Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades”, en este marco de políticas, los retos y metas a cumplir para el año 2021 son las de “incrementar de 4.6 a 5.6 el índice de desarrollo de tecnologías de la información y comunicación”.

A nivel nacional todas las instituciones de servicios públicos priorizaron la implementación de protocolos para evitar ataques cibernéticos ya que en las últimas fechas incrementaron los casos de 20 a 40 millones, estos ataques a estas instituciones hicieron que el Ecuador se ubique en el puesto 31 en la escala mundial de ataques cibernéticos tal como lo indica el Ministerio de Telecomunicaciones [5].

La pandemia del COVID-19, que ha azotado al mundo, en todos los aspectos sociales, económicos y culturales. También tiene sus repercusiones en el campo de la ciberseguridad, la mayoría de ciudades del mundo han sido afectadas por los ciberdelincuentes que han creado nuevas formas de ataques en las redes sociales, en los equipos de hogar y móviles, esto se debe a la decadencia de la crisis sanitaria a nivel mundial, según estadísticas mundiales de acuerdo al informe anual que entrega la INTERPOL [6], en el año 2020 en los meses comprendidos entre enero y abril, se detectaron cerca de 907.000 correos basura, 737.000 de tipo malware y aproximadamente 50.000 URL maliciosas todas relacionadas con la COVID-19.

El informe de esta entidad fue realizado en 194 países afiliados a esta organización, teniendo como resultado final que, en los meses de mayor auge de la pandemia, es decir de enero a abril de 2020, se registró en Europa un total de 48% delitos cibernéticos, en el continente americano el 12%, y el resto del mundo un total del 40%, todos estos delitos relacionados directamente a la pandemia.

Las principales vulnerabilidades que se identificaron y detectaron son: las estafas por *Internet* y *phishing*, esta modalidad se dio a través de correos electrónicos sobre la COVID-19, muy frecuentes con suplantación de identidad un total del 59% de población mundial sufrió este tipo de estafas, también Figura los *malwares disruptivos*, estas intromisiones provocan pérdidas temporales de información esencial en las infraestructuras estatales, según este informe un total del 36%, fue atacado por esta modalidad, los *malwares* de información de datos, los mismos que tratan de recopilar la información personal necesaria para cometer delitos de estafa, del total de países encuestados el 22% sufrió este tipo de ataque en la información personal, finalmente la *desinformación*, que es una nueva modalidad de ataque cibernético con el aumento de la incertidumbre dentro de la población creció de manera exponencial alcanzando un 14%, con teorías de conspiración, se ha dejado un espacio significativo para la práctica de ciberataques.

A nivel de región en Sudamérica la tendencia más clara de ciberdelincuencia relacionada al COVID-19, es la del *malware ransomware* que ataca a pequeñas y medianas empresas, estos ataques tuvieron su nivel máximo en el mes de abril de 2020, es decir que hasta la fecha actual aún se puede tener consecuencias de estos ataques. En el contexto de las redes sociales se aumentó los casos por delitos sexuales, ya que por estos medios pueden localizar con mayor facilidad a sus víctimas por medio del intercambio de imágenes con contenido sexual en menores de edad. Otro de los ataques frecuentes son los dominios maliciosos, relacionados a la COVID-19, un alto grado de estos sitios se utiliza para incidir en acciones malintencionadas, solo en el mes de marzo de 2020 se detectaron cerca de 116.000 nuevos dominios de los que 3.000 resultaron ser maliciosos, y cerca de 40.000 de alto riesgo. Para el mes de junio del mismo año se detectaron 200.000 dominios maliciosos.

La respuesta que esta organización ha efectuado son las reuniones virtuales continuas con las entidades de seguridad que tiene cada país, en las cuales se determina nuevas estrategias para combatir estos casos, la participación continua de foros y debates sobre esta problemática ha facilitado la concientización de cómo actuar ante estas amenazas. La recomendación que esta entidad sigue es mejorar la colaboración entre policía y ciudadanía, creando plataformas colaborativas que ayuden a disminuir la ciberdelincuencia apoyando planes de prevención y sensibilización en la población ayudando de esta manera a mejorar la capacidad y estableciendo nuevas alianzas entre el sector público y privado.

Para tener una estabilidad del software el proceso de aprobación debe abordar la seguridad en cada fase del ciclo de vida del mismo, hay que integrar la seguridad en dos partes, la primera haciendo un seguimiento en los diseños seguros y la segunda incluyendo buenas prácticas seguras. El modelo adecuado está conformado por dos o más modelos, sin embargo, lo importante no es ese modelo sino las técnicas o buenas prácticas de seguridad, como lo sugiere Marulanda y Cevallos [7].

Desde los años 50, el desarrollo de software comenzó como una actividad desordenada a menudo mencionada como código y arreglo. Esto funcionaba bien para sistemas pequeños, pero a medida que los sistemas crecían se hacía más difícil añadir nuevas características y los errores eran más difíciles de arreglar, hasta que se introdujo una alternativa, como lo es la metodología, la misma que propone un proceso sistemático al perfeccionamiento del software haciéndolo más eficaz, Awad, M. A. [8]. Las primeras metodologías llamadas Tradicionales, utilizan como base la obtención y documentación de un conjunto de requisitos, luego de esto se pasa a la elaboración e inspección de diseños arquitectónicos de alto nivel. Estos aspectos, hicieron que los desarrolladores las encontraron frustrantes [8].

Debido a esto nacieron las metodologías de desarrollo ágiles, las mismas que se encuentran contenidas en el Manifiesto Ágil [9], el nombre ágil surgió en 2001, cuando diecisiete de los mejores metodólogos de procesos se reunieron para examinar las futuras tendencias en el progreso del software. Estos métodos aseveran asentar énfasis en las personas, la correlación, el trabajo de software, la contribución entre los usuarios y el cambio, en vez de las herramientas, procesos, los planes y los contratos. Las técnicas originalmente adoptadas por la Alianza Ágil fueron el Desarrollo de Software Adaptativo (ASD)[10], Crystal [11], Sistemas Dinámicos Método de Desarrollo (DSDM) [12], programación extrema (XP) [13], desarrollo impulsado por las características [14] y Scrum [15].

Los métodos utilizados dentro del desarrollo ágil, aceptan cambios constantes, además los documentos técnicos de pruebas y el avance en la funcionalidad que no aporte validez al cliente final, Z. Azham [16], a todo esto se agrega los requisitos no funcionales como la seguridad y la actividades que nos ayuden a verificar los riesgos a los que están sometidos y que se analizan como de baja prioridad y que también son enemigos propios de la agilidad, Graham [17]. Debido a la gradual complejidad de los sistemas de software, los diseñadores tienden a centrarse en el diseño e implementación de requisitos

funcionales. Los requisitos no funcionales como la seguridad a menudo se descuidan [18], entonces, las vulnerabilidades se introducen inevitablemente en cada fase de la metodología que se esté utilizando. En esencia, las vulnerabilidades, son las manifestaciones de errores que violan las políticas de seguridad [19].

Tomando en cuentas estas bases, en este trabajo se propone una metodología de desarrollo ágil de software, la misma que parte desde el escogimiento de las experiencias seguras y ágiles, además de la localización de los ambientes estratégicos probables, con la finalidad de que permita puntualizar las operaciones a emprender para hallar una ruta favorable hacia una posteridad en la implementación de seguridades en las diferentes fases que comprende un ciclo de vida de desarrollo de sistemas.

Los resultados alcanzados por esta investigación han sido favorables en los términos de la propuesta metodológica ya que su validación en un caso práctico dentro de una empresa de desarrollo de software, permitió conocer cuáles son las falencias y los principales problemas que se presentan al momento de la implantación de una nueva metodología, además de conocer cuál es la óptica que el desarrollador tiene en base a los requisitos de seguridad que el usuario final exige para su producto final. la recomendación principal es que para implementar esta metodología se debe tener conocimientos previos a seguridad interna con esto se evitan pérdidas de tiempo innecesarias para el equipo de desarrollo de cada empresa.

Este documento se ha organizado de la siguiente manera. El capítulo 1, contiene el Marco Teórico referencial del trabajo, los antecedentes conceptuales y referenciales, el estado del arte y por último los antecedentes contextuales de la propuesta realizada. El capítulo 2, se describe la metodología y los materiales que utilizaron en la ejecución del presente trabajo, así como también, tipo de estudio, los métodos y técnicas usados para el procesamiento de los datos obtenidos. El capítulo 3, mostrara la obtención de los resultados de acuerdo al estudio realizado; para luego llegar al capítulo 4, el mismo que contiene dos aspectos la primera parte la corroboración de los resultados obtenidos, así como el debate de los mismos resultados, por último, se indican las recomendaciones y conclusiones obtenidas de todo el desarrollo de la investigación.



## PLANTEAMIENTO DEL PROBLEMA

En la actualidad la dimensión de la seguridad es un tema muy importante, los ataques más importantes se enfocan en aplicaciones de software. Una de las fuentes del problema surge a raíz de la hipótesis principal de esta investigación como es el diseño de una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas, entonces se minimizará vulnerabilidades a los ataques informáticos. Por este motivo se hace hincapié en la creación de una metodología, que además de tener su fase de desarrollo del software, cuente con técnicas que faciliten al desarrollador crear estas aplicaciones seguras de acuerdo a las exigencias actuales

La Ingeniería del Software según la definición de la **IEEE** es “la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software”. Por otra parte, autores como Pressman y Maxim [20], la definen como: “las instrucciones que cuando se ejecutan proporcionan características, funciones y rendimiento deseados”, en otras palabras los datos, permiten manipular de forma adecuada la información expresada en forma virtual como impresa.

En la actualidad, el desarrollo de software utiliza modelos seguros, en diferentes fases, sean estos dentro del desarrollo o en la etapa final el despliegue, en algunos casos se sigue aplicando los de la ingeniería de software, estos últimos no consideran la seguridad. Algunos modelos la consideran ya cuando el producto está en producción, se realizan pruebas enfocadas en medir la seguridad del producto. Este problema surge porque las metodologías de la ingeniería de seguridad propuestas, difieren con las que utiliza la ingeniería de software, con lo cual el desarrollador no cumple este cambio y continua con su proceso de desarrollo normal.

En conclusión, los métodos de seguridad que, en la actualidad se han propuesto determinan un grado de complejidad alto para los miembros del equipo de diseño, por lo tanto, se plantea el siguiente problema: **¿Cómo agregar la dimensión de la seguridad durante el desarrollo rápido de software, y presente un mínimo de vulnerabilidades frente ataques informáticos?**

## **OBJETIVOS**

### **Objetivo General:**

- Diseñar una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas, a fin de que permita minimizar vulnerabilidades frente a los ataques informáticos.

### **Objetivos Específicos:**

- Comparar diferentes metodologías de desarrollo ágil de software y ver cuáles son las que incluyen la dimensión de la seguridad
- Diseñar la metodología de desarrollo ágil de software que agregue la dimensión de la seguridad.
- Validar la propuesta metodológica, mediante la aplicación de técnicas de seguridades informáticas, para la minimización de vulnerabilidades frente ataques informáticos.

## **CAPÍTULO 1 MARCO TEÓRICO REFERENCIAL**

En este capítulo, se aborda investigaciones y trabajos de algunos autores que desarrollaron investigaciones encaminadas en buscar la integración de la seguridad con sus proyectos.

### **1.1 ANTECEDENTES HISTÓRICOS DE LA INVESTIGACIÓN**

Como se explicó en líneas anteriores, el objetivo de las metodologías de desarrollo ágil, se fundamentan en la creación de prototipos del proyecto, así mismo van introduciendo mejoras continuamente, es decir, se basan en el funcionamiento iterativo del primer prototipo hasta el final del proyecto. Para el desarrollo de este capítulo, se ha tomado como guía de investigación la propuesta de Kitchenham [21].

#### **1.1.1 Preguntas de Investigación**

Para la presente investigación, se han tomado como referencia las siguientes preguntas, de las cuales delimitaremos la más importante, la cual nos permitirá encontrar la información suficiente y nos guiará en la solución al problema planteado.

- P1. ¿Cómo agregar la dimensión de la seguridad durante el desarrollo de software ágil?
- P2. ¿Cuáles son las vulnerabilidades más comunes en el desarrollo software?
- P3. ¿Cuáles son las técnicas, métodos que aportan seguridad y qué se puedan utilizar en cada fase de la metodología para disminuir los ataques internos y externos?
- P4. ¿Cómo se puede saber cuál es la metodología más segura ante ataques internos y externos?
- P5. ¿Cuál sería la gestión de la seguridad más adecuada en el desarrollo ágil de software?

El SLR, proyecta como base la pregunta de motivación. Para la presente investigación la pregunta que motiva esta revisión es:

- ¿Cómo agregar la dimensión de la seguridad durante el desarrollo rápido de software, y presente un mínimo de vulnerabilidades frente ataques informáticos?

### 1.1.2 Proceso de Búsqueda

Para este proceso de búsqueda se citaron fuentes primarias y secundarias de artículos científicos, específicos desde 2016 hasta la actualidad esta exploración se la realizo de forma manual y se tomaron en cuenta revistas indexadas de las bases de datos, estipuladas en el instructivo de titulación de la UTMACH. Las revistas se seleccionadas incluían estudios empíricos o encuestas bibliográficas, y se utilizaron como fuentes para otras revisiones de literatura relacionadas con el tema en cuestión.

Para realizar la búsqueda se utiliza la base de datos Web de la Ciencia o en inglés como más se la conoce Web of Science, ya que es un servicio en línea de información científica más grandes del mundo, integrado a la web del conocimiento, facilita el acceso a una variedad bases de datos en las que se encuentran artículos de revistas científicas y otros materiales impresos que abarcan los campos del conocimiento académico, la Tabla 1 describe el proceso de búsqueda.

FUENTE	ACRONICO
Acceso al IEEE	IEEE
DIARIO DE APLICACIONES DE RED E INFORMÁTICAS	JNC
REVISTA DE CALIDAD DE SOFTWARE	JSQ
REVISTA INTERNACIONAL DE GESTIÓN DE PROYECTOS DE TECNOLOGÍA DE LA INFORMACIÓN	JIT
CIENCIA DE LA SEGURIDAD	SC
REVISTA DE GESTIÓN DE BASES DE DATOS	JDM
SEGURIDAD DE LA INFORMACIÓN DE LA IET	IET
REVISTA DE GESTIÓN DE INGENIERÍA	JME
REVISTA INTERNACIONAL DE TECNOLOGÍAS Y SISTEMAS DE INFORMACIÓN ENFOQUE	IJITSA
CIENCIA Y TECNOLOGÍA TSINGHUA	ST
REVISTA INTERNACIONAL DE COMPUTACIÓN EN RED Y UTILITARIA	IJGUC
REVISTA DE SISTEMAS Y PROGRAMAS INFORMÁTICOS	JSS
REVISTA INTERNACIONAL DE GESTIÓN DE PROYECTOS	JPM
REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS	GEINTEC
ENCUESTAS DE COMPUTACIÓN ACM	ACM
REVISTA INTERNACIONAL DE SISTEMAS DE INFORMACIÓN EN EL SECTOR DE LOS SERVICIOS	JSI

Tabla 1. Proceso de búsqueda, elaboración propia del autor

### 1.1.3 Criterios de Exclusión Y Inclusión

Dentro de los criterios de EXCLUSIÓN se consideraron los siguientes parámetros:

- Estudios que no se incluyeron en las bases de datos de selección
- Estudios duplicados.
- Artículos de cursos, libros o artículos que sean de estudios primarios
- No buscar resúmenes, entrevistas, noticias

Se han tomado en cuenta los títulos y resúmenes correspondientes en los que se denotarán los criterios siguientes de inclusión:

- Artículos publicados en los últimos 5 años
- Que las fuentes de información y bases de datos, sean las especificadas por la Universidad UTMACH.
- Que los trabajos pertenezcan al área de investigación (seguridades, vulnerabilidades, ciclos de vida, metodologías de desarrollo ágil, *frameworks*)
- Que los artículos sean basados en investigaciones reales.
- Estudios realizados en el campo de seguridades en ciclos de vida de software
- Artículos científicos y documentos de conferencias.

### 1.1.4 Grupos de Control

Para esta sección se ha tomado en cuenta el siguiente criterio con lo que respecta al escogimiento de los estudios primarios se llevan a cabo los siguientes filtros de revisión:

- Filtro de revisión 1:
  - Con relación al Título: se revisan los títulos detalladamente, de las publicaciones arrojadas en las bases de datos establecidas.
  - Resumen: a continuación de los títulos seleccionados, se debe hacer una revisión y lectura del resumen de forma completa.
- Filtro de revisión 2:

Texto Completo: Las publicaciones que pasaron el primer filtro, en sus dos condicionantes, se someten a la lectura y análisis completo.

### 1.1.5 Cadena de Búsqueda

Los artículos, buscados tanto en idioma español como en inglés, utilizaron consultas que incluyen operadores lógicos para maximizar la cobertura de elección de artículos que contengan las palabras claves de las búsquedas. Estas Las cadenas debieron adaptarse en al motor del buscador, ya que no están estandarizados y cada uno cuenta con su sintaxis de consulta como lo indica Kitchenham [21], la Tabla 2, muestra la cadena de búsqueda y sus resultados.

Búsquedas	Bases de datos	Resultados
<b>framework and safety in the life cycle</b>	IEEE	48
	Scholar	4
	WOS	314
<b>agile development framework and methodologies</b>  <b>("agile development framework") and ("methodologies") :&gt;2018</b>	WOS	222
	IEEE	206
	Scholar	53
<b>(Agile development methodologies) AND (Latest generation agile methodologies)</b>	Scholar	17.200
	WOS	1
<b>Computer security and agile development methodologies</b>	WOS	3
	Scholar	16.900
	IEEE	32
<b>vulnerabilities and agile methodologies ("vulnerabilities") and ("agile methodologies") :&gt;2018</b>	IEEE	9
	Scholar	287
	WOS	8
<b>assurances in agile methodologies</b>	WOS	17
	IEEE	53
<b>security attribute and software vulnerabilities</b>	WOS	38
	IEEE	112
	Scholar	16.900
<b>information security and software life cycles or agile development</b>	IEEE	8
	Scholar	15.800
	WOS	2.609
<b>métricas e indicadores de gestión de incidentes de seguridad de la información</b>	IEEE	1
	Scholar	16.000

Tabla 2. Cadena de búsqueda y sus resultados, elaboración propia del autor

Realizada la búsqueda, utilizando las cadenas indicadas, los resultados que arrojaron las mismas obedecen a cinco bases de datos reconocidas por el instructivo:

- IEEE Explorer
- WOS
- Springer
- IGI – Global
- Elsevier

### 1.1.6 Selección de Estudios

Para esta sección, la selección de criterios, predestinados a reconocer los estudios primarios, que brindan respuesta directa e inmediata sobre la pregunta que aborda la investigación, tiene por finalidad reducir el sesgo. Además, los criterios de inclusión y exclusión se basan en la pregunta de investigación.

La Tabla 3, describe la lectura y clasificación de los artículos que cumplen los criterios de inclusión [20].

Bases de Datos	Artículos
IEEE Explorer	469
WOS	584
Springer	25
IGI – Global	18
Elsevier	23

Tabla 3. Artículos que cumplen los criterios de inclusión, elaboración propia del autor

### 1.1.7 Resultados de la Revisión

Luego de haber realizado el SLR, se hallaron 80 artículos, luego se filtró en 35 los cuales estaban más cerca al objetivo planteado para esta investigación como es *Diseñar una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas, a fin de que nos permita minimizar vulnerabilidades a los ataques informáticos*, estos artículos cumplen las restrictivas planeadas para la *revisión sistema*. Las fechas de publicación de estos artículos,

están en el intervalo de tiempo establecido; esto es, 2016-2020. Los 45 artículos restantes quedan descartados por no cumplir los criterios de inclusión establecidos para esta investigación.

Ahora revisamos algunos estudios y la manera de como abordaron los autores las diferentes problemáticas que se plantearon para resolver sus investigaciones.

En el trabajo realizado por Altunel [22], muestra la evolución de las metodologías ágiles. El propósito que tienen estas y los cambios que se han dado en el transcurso del desarrollo. Este trabajo consta de la revisión que se hizo a cada metodología ágil, desde la perspectiva de su ciclo de vida, así como la referencia al estudio realizado en la relación que existe entre el proyecto y el producto.

El trabajo de Buchalcevova [23], detalla a fondo una revisión completa de todas las metodologías de desarrollo. El objetivo principal del artículo se basa en la examinación de los marcos de trabajo escalado, seleccionándolos y comparándolos entre metodologías, el sistema desarrollado por ellos lo utilizan para la evaluación y comparación, las conclusiones de este artículo son que las grandes empresas pueden hacer el uso de estas metodologías de desarrollo.

Algunos autores enfocan la seguridad como un atributo poco usado en cada fase del ciclo de vida del software, los trabajos realizados se enfocan en pruebas de penetración, experiencias de usuarios, definición de prácticas para controlar vulnerabilidades específicas que se dan en las fases de diseño y desarrollo.

Para Casola [18], nos muestra que la evaluación de la seguridad es una actividad costosa y que requiere demasiado tiempo en su aplicación, es por ello que su trabajo se basa en la aplicación de pruebas de penetración a nivel de aplicación dentro de la representación de la arquitectura de la aplicación, este proceso se puede aplicar fácilmente en el proceso continuo de desarrollo de integración y así facilitar a los desarrolladores la evaluación de la seguridad del software.

En otro artículo relevante es el realizado por, Gu; [24], que tiene como nombre Un enfoque para analizar la vulnerabilidad del flujo de información en la arquitectura de software, traducido al español, en el año 2020, se muestra la seguridad de una forma diferente ya que el método desarrollado solo se puede aplicar una vez que software esté completamente terminado, cuando los códigos fuente estén disponibles. En este trabajo se propone un análisis de vulnerabilidades desde el flujo de información, para el investigador la parte



principal es la arquitectura ya que es ahí donde se determina la calidad del mismo y se pueden corregir las vulnerabilidades en una fase temprana del ciclo de vida cuando la revisión es más fácil y con un costo bajo. La propuesta realizada propone un método para construir diagramas de invocación de servicios basados en la teoría de grafos, que puede representar el flujo de información en la arquitectura de software.

Otros desarrolladores como Shah, et al [25], centran sus investigaciones en las vulnerabilidades, este artículo se publicó en la revista The Journal of Defense Modeling and Simulation, en el año 2019, el mismo que propone un análisis completo a las vulnerabilidades partiendo de dos enfoques: el primero utilizando la optimización del valor del atributo individual y el segundo utilizando la optimización del valor del atributo múltiple, el primero presenta una metodología que optimiza la selección de vulnerabilidades para la mitigación con respecto a un atributo individual, el segundo considera múltiples atributos en la toma de decisiones de selección de vulnerabilidades los resultados esperados fueron cotejados por medio de un algoritmo de priorización de vulnerabilidades llamado VULCON.

Para algunos desarrolladores como Jafary & Rasoolzadegan [26], los patrones de seguridad son un medio para encapsular y comunicar soluciones de seguridad probadas. Estos enfoques son establecidos para integrar la seguridad. Sin embargo, la estructura de la solución del patrón elegido se integrará con el diseño de software y, por lo tanto, afectan a muchos atributos de calidad como la flexibilidad y la seguridad. En este trabajo que tiene como nombre "Mutaciones de patrones de seguridad centrados en la calidad", los autores, proponen el concepto de mutaciones de patrones de seguridad centrados en la calidad que se crean mutando los patrones de corriente usando reglas de refactorización de diseño, la metodología propuesta tiene tres partes, en la primera aumentan el patrón añadiendo los atributos y métodos mínimos a la clase pero no se añaden funciones o atributos secundarios, en el segunda fase los desarrolladores agregan un procedimiento llamado el UMLSec, esto con el fin de medir las métricas pertinentes de seguridad y en la tercera fase se realiza la evaluación del mismo.

En el trabajo realizado por, Yao, et al [27], en el cual nos muestra un sistema desarrollado por ellos que les permite, realizar un análisis profundo de principio a fin y así ir detectando los ataques en el programa revisado, para la ejecución de este trabajo se realizaron varias pruebas de penetración de software y cada una arrojó resultados positivos y desconocidos para los desarrolladores, pero su óptica de seguridad solo va enfocada a la codificación de los programas; es decir se emplea solamente cuando el sistema está terminado totalmente.

## **1.2 ANTECEDENTES CONCEPTUALES**

### **1.2.1 Hipótesis de la Investigación**

- Si se diseña una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas, entonces se minimizará vulnerabilidades a los ataques informáticos.

### **1.2.2 Categorización de las Variables**

#### **1.2.2.1 Variable Dependiente:**

- Minimización de vulnerabilidades del software frente ataques informáticos

#### **1.2.2.2 Variable Independiente:**

- Diseño de una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas.

### **1.2.3 Red de Categorías**

La Figura 1, representa la red de categorías de las variables de la investigación. Esto siguiendo el proceso de investigación que sugiere Hernández Sampieri [28].

## Red de categorías

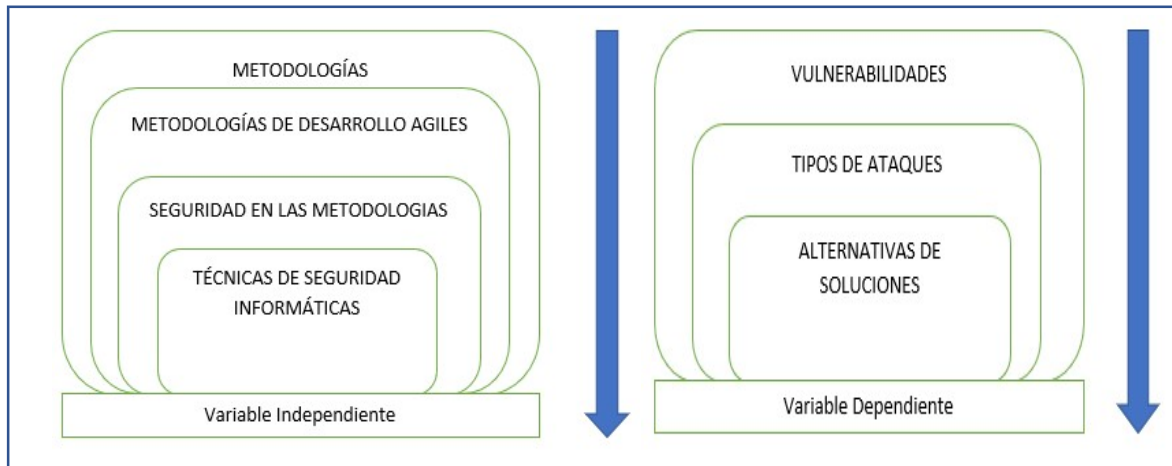


Figura 1. Red de categorías de las variables de investigación, elaboración propia del autor

### **1.2.4 Fundamentación Teórica de la Variable Independiente**

Es necesario abordar la conceptualización de como varios autores, desarrolladores e investigadores conceptualizan los términos como: seguridad, vulnerabilidades y otros conceptos que se toman en cuenta para el presente proyecto planteado.

#### **1.2.4.1 Metodologías**

Según el libro la Metodología de la Investigación, la metodología es un conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que rige una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos ocuidados específicos. Con frecuencia puede definirse la metodología como el estudio o elección de un método pertinente o adecuadamente aplicable a determinado objeto. Así mismo no se debe llamar metodología a cualquier procedimiento, pues se trata de un concepto que en la gran mayoría de los casos resulta demasiado amplio, siendo preferible usar la palabra método [29].

En el trabajo realizado por Ríos et al. (2017), [30], aborda la metodología de la siguiente forma son un conjunto pasos estratégicos y organizados con el único fin de obtener el material documental, analizarlo y someterlo a un proceso estricto de revisión, descripción y

reseña. Este análisis se lo realiza de forma comparativa, elaborando las conclusiones y resultados obtenidos.

Para Hernández Sampieri la metodología es una serie de pasos ordenados a través de técnicas y métodos de inflexibilidad científica, que se utilizan en proceso de investigación, para de esta manera alcanzar resultados teóricamente demostrados en el sentido en que se haya aplicado la investigación. Por tal motivo se dice que la metodología se la utiliza como un puntal conceptual que rige la forma de cómo se vayan aplicar los procedimientos dentro de una investigación. Para este autor la metodología se divide en dos la *cuantitativa* y la *cualitativa*, la primera, es aquella que se la emplea el uso de las ciencias naturales ya que sus datos obtenidos son cuantificables permitiendo de esta manera el acceso a los mismos y así poder medirlos para el uso que se les aplique, ya sea utilizando alguna técnica como la estadística ya que son datos deductivos que se los puede revisar en una muestra dentro una población determinada.

La metodología cualitativa, en cambio se encuentra en las ciencias sociales, por tal motivo sus datos obtenidos no se los puede cuantificar. Esta particularidad hace que estos valores obtenidos se los demuestre subjetivamente, por medio de herramientas como la entrevista ya que su razonamiento es inductivo.

#### **1.2.4.2 Metodologías de Desarrollo Ágil**

En el manifiesto ágil [9], la filosofía detrás de los métodos ágiles se refleja, poniendo el énfasis en cuatro aspectos clave: la importancia de los equipos con organización propia que tienen el control sobre el trabajo que realizan, la comunicación y colaboración entre los miembros del equipo y entre los profesionales y sus clientes, el reconocimiento de que el cambio representa una oportunidad y la insistencia en la entrega rápida de software que satisfaga al consumidor.

Para los autores, Balijepally, et al (2017) [31], definen a las metodologías de desarrollo ágil, como una orientación incremental evolutivo, que se realiza de manera colaborativa por grupos autoorganizados dentro del marco de gobierno efectivo, la cual produce un alto grado de soluciones de calidad de manera rentable y oportuna que satisface las necesidades cambiantes de sus partes interesadas.

En el trabajo de Ríos et al. (2017), [32], las metodologías ágiles son fácilmente modificables por el equipo desarrollador de sistemas, la característica principal es permitir la subdivisión del mismo proyecto en pequeñas fracciones permitiendo así establecer periodos de tiempo cortos agilizando las entregas en cada etapa, los cambios que el usuario final aplique en el transcurso del ciclo de vida, se adaptan fácilmente a la estructura que la misma metodología sugiere, de la misma forma se da una garantía total del producto final.

En el trabajo desarrollado por de Freitas, et al. (2019) [33] , se analizan dos metodologías de desarrollo ágil de forma comparativa, con el único fin de valorar los resultados obtenidos y ponderar el nivel alcanzado en estas áreas de la ciencia. Estos autores realizan un análisis mediante la herramienta casos de uso aplicados en un determinado proyecto de desarrollo de software, obteniendo de esta forma los riesgos que produce la implementación de estas metodologías, ya que para cada tipo de proyecto se debe implantar un enfoque diferente obteniendo resultados actuales y flexibles en el transcurso del ciclo de vida del software

#### **1.2.4.3 Seguridad en las Metodologías**

Algunos autores como Matamoros en el 2018 [34], relacionan a la seguridad con la funcionalidad y la experiencia de usuario. Esta relación tiene un carácter controvertido para los usuarios en los recursos o en el tiempo, ya que produce inconvenientes a la hora de implantar un nuevo servicio. Cada vez que se hace énfasis en uno de los aspectos relacionados los otros dos se van distanciando. Otros trabajos como el de Castellaro, et al. [35], nos dice que la seguridad mantiene relación directa con la confidencialidad, ya que se refiere a los niveles de seguridad que se da para cierto tipo de información, debe tener Integridad, esta propiedad se debe mantener durante todo el desarrollo y la ejecución del software, debe estar disponible a sus usuarios, tener trazabilidad y no repudio, a fin que se pueda tener responsabilidad de las acciones que se hayan ejecutado.

Algunos desarrolladores como G. McGraw [36], enmarcan la seguridad como una Imagen de la Ingeniería de Software que trata que los productos desarrollados funcionen correctamente a ataques maliciosos, sean estos internos o externos. Se resalta que la seguridad no se la puede incorporar en cualquier momento, además el diseño debe desarrollarse en relación a la seguridad, condicionalmente, no será posible incorporarla exitosamente una vez desarrollado el producto. Otra revisión que el autor Ramírez Aguilera, J. A. (2020) [37], nos explica, que la seguridad es la aplicación de principios de la

información en el desarrollo de software, se hace referencia a la protección de la información contra el acceso o a la alteración de la misma.

algunas metodologías como Security Requirements Engineering Procesos, El SREP [38], es un método basado en activos y orientado a riesgos, que permite el establecimiento de requisitos de seguridad durante el desarrollo de las aplicaciones. Básicamente lo que define este método es la implementación de *Common Criteria* durante todas las fases del desarrollo de software. El CC es un estándar internacional ISO/IEC 15408 para seguridad informática, cuyo objetivo es definir requisitos seguros que les permitan a los desarrolladores especificar atributos de seguridad y evaluar que dichos productos si cumplan con su cometido. Esta metodología trata cada fase del Ciclo de Vida del software, como a un mini proceso o iteración, dentro de las cuales se aplican las actividades SREP que permiten identificar y mantener actualizados los requisitos de seguridad de la fase, permitiendo mitigar efectivamente los riesgos asociados a cada una.

El modelo SQUARE, Security Quality Requirements Engineering [39], propone varios pasos para construir modelos de seguridad desde las etapas tempranas del ciclo de vida del software. En el proceso del modelo se hace un análisis enfocado a la seguridad, los patrones de ataque, las amenazas y las vulnerabilidades y se desarrollan malos casos de uso/abuso.

La metodología de desarrollo UMLSec [40], basada en UML, especifica requisitos de seguridad relacionados con integridad y confidencialidad. Mediante mecanismos ligeros de extensión de UML es posible expresar los estereotipos, las etiquetas, las restricciones y el comportamiento de un subsistema en presencia de un ataque. El modelo define un conjunto de operaciones que puede efectuar un atacante a un estereotipo y trata de modelar la actuación del subsistema en su presencia. La metodología UMLSec pretende reutilizar los diseños ya existentes, aplicando patrones para formar nuevos a partir de una transformación al existente.

En resumen, se puede distinguir que, en todos estos trabajos de investigación, la dimensión de la seguridad no ha sido abordada con detalle en cada una de las fases de las metodologías estudiadas, en algunas esta dimensión se enfoca en ciertas fases como el diseño y el desarrollo o simplemente se da seguridad por medio de pruebas a las que se somete el producto final. En esta investigación el objetivo principal, es enfocar a la seguridad como una dimensión esencial que debe estar presente en cada una de las fases

del ciclo de vida del software, esta dimensión no solo debe estar encapsulada a la confidencialidad, sino que debe tener estricta relación con la integridad y la disponibilidad solo así se puede garantizar que el producto final este acorde a las necesidades que demanda la tecnología actual, la Figura 2 muestra gráficamente como se propone esta dimensión.



*Figura 2. Propuesta de la dimensión de la seguridad, elaboración propia*

#### **1.2.4.4 Técnicas de Seguridades Informáticas**

Decimos que la seguridad es la ausencia de riesgo, confianza en algo o alguien, entonces las seguridades informáticas son técnicas desarrolladas para proteger equipos informáticos individuales o conectados a una red frente a daños accidentales o intencionales. Estos daños incluyen el mal funcionamiento de hardware o software.

Según la norma ISO-27000 [41] , los principios de la seguridad informática son: Confidencialidad, integridad, disponibilidad, de acuerdo a estos tres conceptos tenemos dos tipos de seguridad la personal, que se da a nivel de hackers o cracker y la lógica que se da por medio de controles de acceso, autenticación, encriptación, entre otras.

McGraw [42], enfatiza que la seguridad no es un aspecto que pueda ser incorporado a un sistema en cualquier instante, esto implica que, al diseñar un producto de software, este

diseño debe realizarse pensando en el aspecto de seguridad, dado que, una vez desarrollado el producto, no será posible incorporarla exitosamente. Interpretando lo anterior existen algunas técnicas que se pueden adaptar a la forma en la que se desarrolla el software, entre los más relevantes se encuentran:

- Revisión de código utilizando herramientas de análisis estático
- Análisis de riesgo arquitectónico
- Pruebas de penetración
- Pruebas de seguridad
- Desarrollo de casos de abuso

Otros autores como Alenezi y Agrawal Alka [43], sugieren que, para minimizar las vulnerabilidades y los posibles ataques, se debería plantear el uso de tácticas de diseño de seguridad, con el único de poder satisfacer los requisitos de seguridad planteados en un inicio por parte del usuario final y el desarrollador del sistema. Estas tácticas permitirán detectar las anomalías y definir a su vez el factor primordial que ayudará a los arquitectos a definir y asegurar los sistemas.

El trabajo de Ali y Hafeez [44], sugiere priorizar los casos de prueba, a fin de poder mejorar la calidad de las versiones del sistema. Para esto se debe mantener los casos que frecuentemente cambian su estado, en el caso de ser similares, se priorizan en función de su cobertura o aplicabilidad dentro del sistema desarrollado, en el caso de tener muchos fallos la propuesta sugiere validarlos de acuerdo al tema que vayan enmarcados dentro de la aplicación final.

## **1.2.5 Fundamentación teórica de la variable dependiente**

### **1.2.5.1 Vulnerabilidades**

Las vulnerabilidades, son aquellos fallos o agujeros de seguridad, mediante los cuales se pueden ejecutar los riesgos anteriormente comentados. Existen multitud de vulnerabilidades que por diferentes causas exponen a los sistemas y a las aplicaciones a multitud de ataques. Para comprender más detalladamente las causas raíces de estas nos basaremos en la clasificación que realiza sobre las mismas el proyecto OWASP [45].

En algunos trabajos como el de Sánchez, et ál..(2020) [46], nos dice que las vulnerabilidad en el software son un fallo en el sistema, el mismo que permite a los atacantes perpetuar



las medidas de seguridad, se debe tomar en cuenta que los errores y fallos no son nuevos en el mundo de la informática, cualquier sistema sea grande o pequeño podría contener un gran número de fallas en su seguridad, estos errores son explotados por personas malintencionadas que a su vez producen daños u obtienen beneficios de los mismos.

Algunos autores como Akram y Luo, [47], en su trabajo, evalúan las vulnerabilidades a nivel de líneas de código fuente, mediante técnicas de código parches, a nivel de metadatos de archivos, este aporte permite a los expertos en seguridad, revisar los diferentes tipos de vulnerabilidades existentes en el código fuente de un sistema y así poder definir la jerarquía que tienen las mismas. La creación de esta meta data permitirá que otros sistemas puedan identificar las vulnerabilidades en menor tiempo posible evitando de esta manera costos innecesarios al momento del despliegue del mismo producto.

En el trabajo de Gaik-Yee et al. (2016) [48], se presenta un sistema que permite detectar y prevenir las vulnerabilidades a través de reglas y patrones asociados entre sí. De esta manera la detección se hace en el menor tiempo posible evitando así que el intruso se expanda por todo el sistema. Los resultados de este método han permitido prevenir y predecir los futuros ataques dentro de varias plataformas o sistemas de desarrollo, el trabajo a futuro será medir la calidad de software y la medición de los servicios con respecto a la latencia del sistema.

En el trabajo realizado por de Mohino, Juan et al. (2019) [49], se prioriza las vulnerabilidades en cada una de las fases de ciclo de vida que tiene cada metodología, mientras más interacciones se tenga con la integración de los componentes, más rápido se podrá controlar estas vulnerabilidades. Para lograr esto se debe tomar muy en cuenta los entregables que se tenga en cada fase, disminuyendo de esta forma los costos innecesarios y los tiempos extras en la entrega del producto final, solos así se obtiene un producto de calidad y a su vez seguro alcanzando con éxito la funcionalidad total del proyecto de desarrollo de software.

#### **1.2.5.2 Tipos de Ataques**

El proyecto OWASP Top Ten (OWASP, 2020) [45], es un potente documento que analiza y clasifica las vulnerabilidades más recurrentes en el mundo de la seguridad dentro del marco de las aplicaciones web. Según el estudio, las 10 vulnerabilidades que mayor impacto tienen sobre las aplicaciones se pueden agrupar dentro de las siguientes categorías:

- Inyección
- Autenticación rota
- Exposición de datos sensibles
- Entidades externas XML (XXE)
- Control de acceso roto
- Fallas de seguridad y configuraciones
- Escritura de Sitios Cruzados XSS
- Deserialización insegura
- Uso de componentes con vulnerabilidades conocidas
- Insuficiente registro y vigilancia

### **1.2.5.3 Alternativas de Solución**

Para mitigar los problemas que generan las vulnerabilidades y ataques continuos dentro del desarrollo de software ágil varios autores proponen alternativas de solución viables que permitan minimizar estas intromisiones ya que las mismas generan costos innecesarios y tiempo extra al ya marcado en el cronograma de ejecución del proyecto. En esta sección se analizarán algunos trabajos que indican las soluciones que se han tomado y han sido validados.

La investigación realizada por Fahad, Muhammad et al. (2017) [50], proponen la creación de una metodología hibrida, como alternativa para solucionar los problemas de seguridad que afectan a los proyectos de desarrollo de software. Esta nueva tendencia nace a partir de la comparación de varias metodologías ágiles se crea una fusión dando como resultado una nueva metodología que implica las características especiales y fases de las siguientes metodologías de desarrollo ágil como son XP, Scrum y DSDM.

En otra investigación realizada por Seyed y Reza [51], introducen como alternativa de solución las técnicas de aprendizaje automático y minería de datos, para de esta manera amenorar las vulnerabilidades. El trabajo incluye una revisión sistemática de diferentes enfoques en varios campos del análisis que abordan esta cuestión de seguridad, se determinan las ventajas y desventajas que se puede tener al utilizar estas técnicas y se pondera las mejores opciones para obtener mejores resultados que brinden mayores ventajas a la hora de contrarrestar estos ataques.

En el trabajo de Sri Nikhil Gupta et al. (2020) [52], se enfoca a la creación de un marco de mitigación de vulnerabilidades, este marco tiene como propósito principal aplicarse en

cualquier proyecto de desarrollo de software, ya que los que existen actualmente solo se utilizan dentro del marco jurídico y en empresas avaladas por el organismo máximo que rige en cada país. El marco desarrollado contiene una arquitectura adaptable y detallada, priorizando y mejorando las brechas que la ciberseguridad no proporciona a la hora de establecer seguridad en el punto máximo de madurez del proyecto desarrollado.

### **1.3 ANTECEDENTES CONTEXTUALES DE LA INVESTIGACIÓN**

#### **1.3.1 Delimitación del Contexto de Estudio**

Como hemos revisado en la sección anterior, los trabajos desarrollados por grandes compañías dedicadas al desarrollo de proyectos de software, así como también las investigaciones realizadas individualmente por desarrolladores, nos dicen que las metodologías ágiles están ganando una gran notoriedad en su campo, el termino Proyecto es una de las palabras más utilizadas en la actualidad, usada en casi todos los negocios del siglo XXI, obviamente con sus respectivos alcances y percepciones. De este término nace otra definición llamada ciclo de vida del software, el mismo que se conceptualiza como un conjunto de fases a seguir desde que inicia hasta llegar a su cierre completo.

Dentro de estas fases se encuentran diferentes procesos, para poder ejecutarlos y cumplir con la presentación de los documentos pertinentes que resulten en el proceso, estas fases tienen sus respectivas técnicas y métodos, a su vez estos métodos se relacionan con los atributos que le dan una mejor efectividad a cada fase del ciclo de vida, para el planteamiento de este proyecto se ha tomado como referencia el atributo de seguridad, ya que después de haber revisado detalladamente, los trabajos realizados se demuestra que este atributo no se lo está aplicando como debería ser, por tal motivo es un problema a investigar y al cual se le puede agregar una solución acertada y que se pueda tomar como referencia para futuras investigaciones.

#### **1.3.2 Propuesta de Solución y Contribuciones**

Después de haber revisado el concepto y la forma de trabajo de las metodologías de desarrollo ágil podemos denotar la importancia que tienen para esta propuesta ya que sus características básicas están basadas en el desarrollo incremental e iterativo, es decir, que las soluciones van a ir evolucionando según la necesidad que tenga el proyecto, el trabajo va ser colaborativo entre los equipos que estén involucrados en el proyecto. Dentro de esto también se observa que todas cuentan con ciclo de vida básico y la mayoría consisten en

las siguientes fases como son: la planificación, análisis de requisitos, diseño, desarrollo y pruebas.

La seguridad dentro de las fases que tienen las metodologías de desarrollo ágil, se la enfoca desde diferentes ópticas algunos sugieren aplicar métodos cuando el software está terminado, otros aplican técnicas para minimizar las vulnerabilidades en etapas claves como son la arquitectura y el desarrollo, el enfoque que tomamos para esta propuesta, va desde el inicio hasta el cierre de la misma.

Algunas metodologías de desarrollo ágil, utilizan marcos de trabajo o *framework* adaptables a su ciclo de vida. El objetivo de esta investigación es proponer una metodología de desarrollo ágil de software, con la particularidad de agregar el atributo de seguridad, para lo cual nos basaremos en las técnicas y métodos más convenientes, para de esta forma introducirla en cada fase y así tener menor vulnerabilidad ante ataques informáticos, cumpliendo de esta manera con el alcance total de esta investigación. Para mayor comprensión de la propuesta que se quiere desarrollar mostramos la siguiente imagen, del ciclo de vida que comprende la propuesta metodológica, en la Figura 3.



Figura 3. Ciclo de vida propuesta metodológica LG-SEG

Como se puede ver en la gráfica lo que se pretende es incorporar un método o técnica que permita detectar las vulnerabilidades desde la primera fase del ciclo de vida con esto

evitaremos perdidas de tiempos y gastos innecesarios ya que al detectarlos desde un principio se pueden corregir en el tiempo destinado para su ejecución.

### **1.3.3 Organización del Documento**

El documento está organizado con base en cuatro capítulos: la parte de la Introducción, que es el componente inicial que contextualiza y sintetiza desarrollado en los capítulos aquí se facilita la comprensión y significancia del trabajo para el lector. En el capítulo 1, se expone la visión general de la metodología de la revisión sistemática de la literatura, comenzando por los antecedentes históricos de la investigación, pasando luego a los antecedentes conceptuales y finalizando en los antecedentes contextuales de la misma.

En el capítulo 2, describe la metodología y los materiales usados en la realización del trabajo. En este capítulo se describe y explica: el tipo de estudio o investigación realizada, el paradigma o enfoque desde el cual se realizó, la población y la muestra, los métodos teóricos con los materiales utilizados, los métodos empíricos con los materiales utilizados y finalmente las técnicas estadísticas para el procesamiento de los datos obtenidos.

El capítulo 3, describe los resultados obtenidos del estudio realizado. Aquí se detalla la fundamentación del aporte práctico y su elaboración, contiene dos aspectos, primero la fundamentación teórica de la propuesta metodológica y en segundo lugar la propia propuesta metodológica.

Por su parte, en el capítulo 4, se describe la discusión de los resultados en el estudio realizado y su corroboración o validación, sea esta de forma cualitativa, cuantitativa o mixta. Finalmente se detallan las conclusiones y resultados obtenidos de toda la investigación realizada, así como su bibliografía como base de referencia de la búsqueda de la información.

## CAPÍTULO 2 MATERIALES Y MÉTODOS

En este capítulo, se describe la metodología y los materiales que se utilizaron en la realización de la investigación, así como el enfoque en el cual se realizó la misma, también veremos cómo se calculó la muestra dentro de la población de estudio, los métodos teóricos y empíricos utilizados, y finalmente las técnicas que se usaron en el proceso de obtención de la información para definir los resultados que defiende. Esto como evidencia de la competencia del estudiante.

### 2.1 Tipo de Estudio o Investigación Realizada

En líneas anteriores se realizó la revisión de la literatura y se afinó el planteamiento del problema, se consideraron los alcances, iniciales y finales, que tiene esta investigación, en términos de conocimiento, hasta dónde es posible que llegue este estudio. Para ello se ha determinado que esta investigación sea de tipo Exploratoria, ya que el objetivo de la misma es: *Diseñar una propuesta metodológica de desarrollo ágil de software que agregue el atributo de seguridad, mediante la aplicación de técnicas de seguridades informáticas, a fin de que nos permita minimizar vulnerabilidad a los ataques informáticos*, con la revisión de trabajos realizados por otros autores podemos determinar que en este campo no se han realizado estudios de este tipo. La Figura 4, indica el alcance y tipo de investigación.

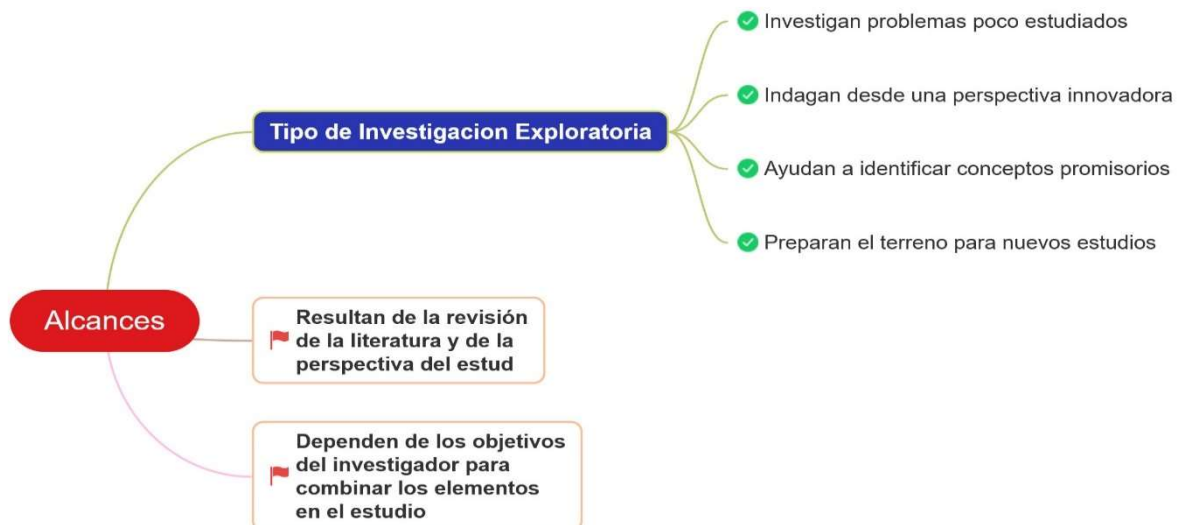


Figura 4. Alcance y Tipo de la Investigación, elaboración propia

## 2.2 Paradigma o enfoque en el cual se realizó

El enfoque de la investigación, se ha establecido que sea **cuantitativo**. ya que mediante este proceso permite recolectar la información para probar la hipótesis, la base que se utiliza será la medición numérica y el análisis de tipo estadístico, para establecer patrones de comportamiento y comprobar la teoría. Este enfoque permitió seguir una secuencia ordenada de pasos como: partir desde una idea general, luego se han derivado las preguntas de investigación, revisión de la literatura y la construcción del marco teórico. Después de esto se determinó la hipótesis y se determinan las variables. Este proceso se muestra en la Figura 5:

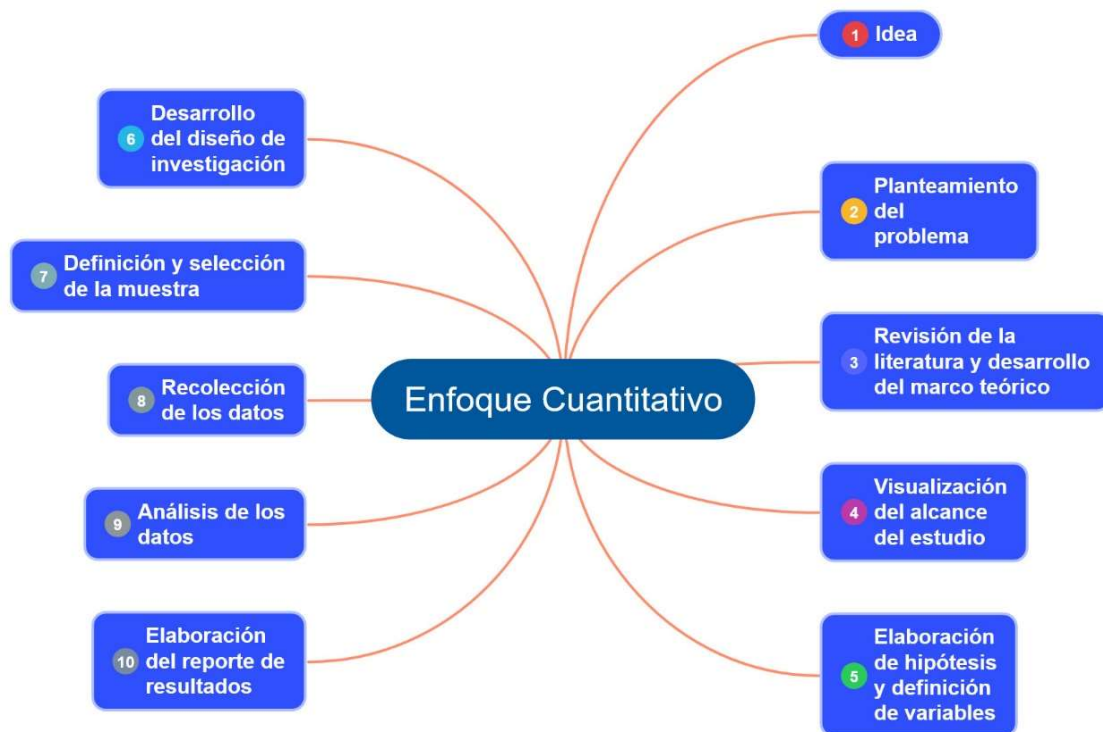


Figura 5. Enfoque Cuantitativo y los pasos que contiene, elaboración propia

El plan de acción que indica la secuencia de pasos a seguir, es el diseño de la investigación su propósito principal, es dar solución a las preguntas de investigación. Así mismo cumplir con los objetivos, es por eso que el diseño de esta investigación será de tipo **cuasiexperimental**. En este diseño, la asignación de sujetos a la variable independiente, no se realiza de forma aleatoria, sino que la selección es a grupos de individuos que ya están formados. Dado que esta investigación es la creación de una propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, el grupo al que va dirigido será el departamento de desarrollo, de la empresa SOLNUS de la ciudad de Loja.

La principal característica que nos brinda este diseño, es que la variable independiente se puede manipular. Esta característica se comparte en los diseños cuasiexperimentales y los experimentales. Ambos diseños tienen como objetivo el estudio de la causa - efecto entre la variable independiente sobre la dependiente. Para mayor comprensión la Figura 6 muestra la teoría del diseño cuasiexperimental, además en la Tabla 4, describe la variable causa y la variable efecto.

CAUSA		EFECTO	
VARIABLE INDEPENDIENTE		VARIABLE DEPENDIENTE	
Diseño de una propuesta metodológica de desarrollo de software ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas		Minimización de vulnerabilidades del software frente ataques informáticos	

*Tabla 4. Variables Causa - Efecto, elaboración propia del autor*



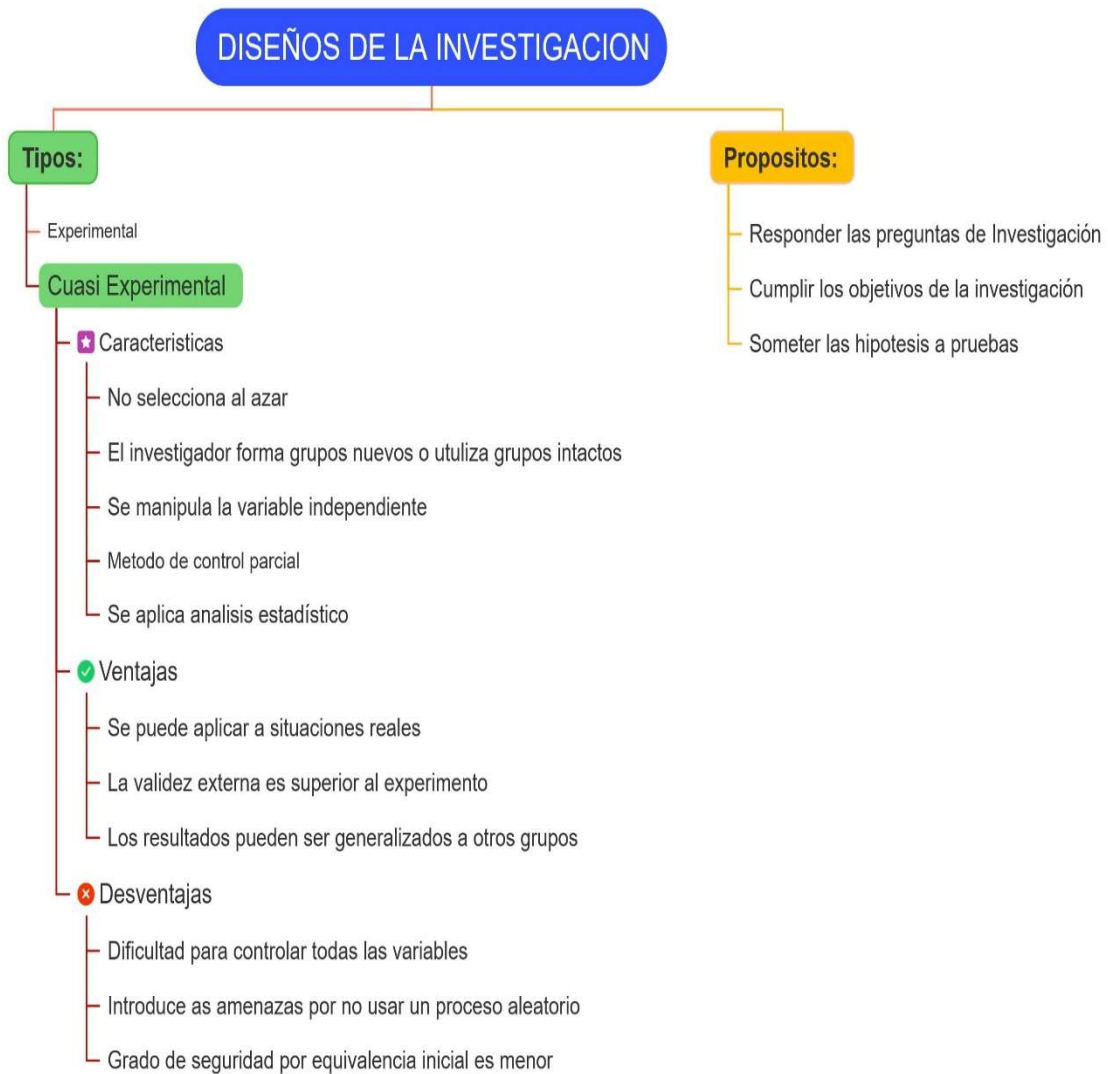


Figura 6. Diseño de la Investigación – Tipos - Cuasi-Experimental – Características, Ventajas y Desventajas, elaboración propia

### 2.3 Cálculo de la Población y Muestra

Dentro del proceso de selección, la muestra, es un subgrupo de la población, de interés en el cual se recogerán los datos, el mismo que debe delimitarse y definirse, con precisión. Para esta sección se ha tomado en cuenta como población las empresas que desarrollan software, el autor de esta investigación reside en la ciudad de Loja, por este motivo se ha tomado en cuenta las empresas que existen en esta localidad, teniendo como resultado que cinco empresas del medio desarrollan software. Para esta investigación la muestra será

la empresa SOLNUS de la ciudad de Loja, ya que ha permitido desarrollar con facilidad las pruebas que conllevan la implementación de una nueva metodología, esta empresa en la actualidad cuenta con el siguiente mapa jerárquico, la Figura 7, muestra el Organigrama funcional empresa SOLNUS.

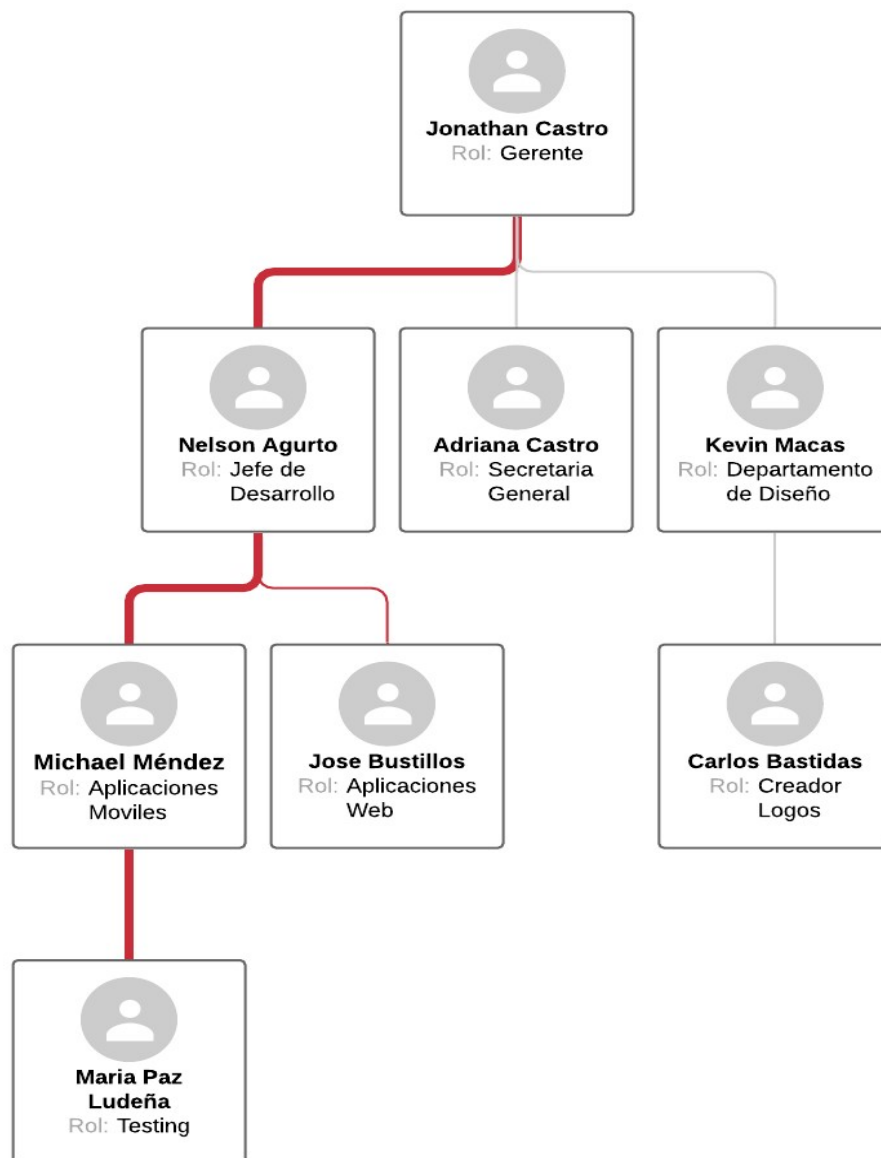


Figura 7. Organigrama funcional actual de la Empresa SOLNUS, ciudad de Loja, elaboración propia del autor

La línea roja que se muestra en la Figura 7, es el departamento en el cual, se aplica la técnica que nos permita recolectar los datos necesarios para validar nuestra teoría.

Teniendo como población limitada al departamento de desarrollo, tal como lo indicamos en la sección anterior, se utiliza el diseño cuasiexperimental, la técnica a utilizar será implementada en un grupo ya existente, cuando la población es limitada la muestra es igual a la población, como nos muestra la Figura 8:

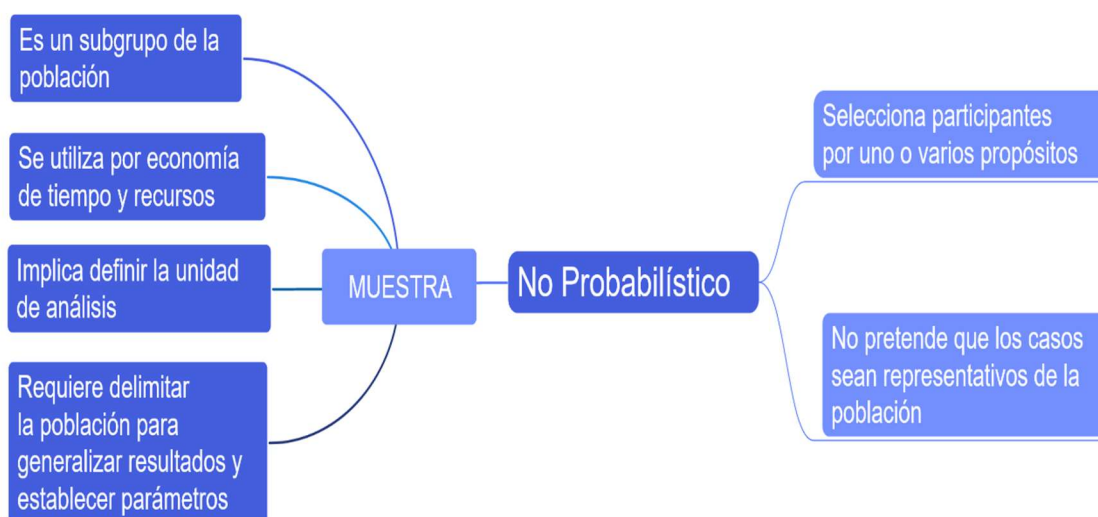


Figura 8. Esquema del tipo de Muestra, elaboración propia del autor

En este caso la muestra es no probabilística, ya que está dirigida a un grupo existente. Su procedimiento es de selección informal, se manipulan en diversas investigaciones cuantitativas o cualitativas. Estas muestras dirigidas implican algunas desventajas, la principal es no poder realizar un cálculo con precisión con relación al error estándar, de esta forma, no se puede calcular el nivel de confianza se está estimando con los datos. La principal ventaja que tiene este tipo de muestra es su utilidad para determinado estudio, ya que no requiere representatividad sino una controlada elección de casos previos al planteamiento del problema.

## 2.4 Métodos Teóricos con los materiales utilizados

Para esta recolección se utilizó el método de la **entrevista**, ya que, está dirigida hacia un grupo formado y no de forma aleatoria, esta técnica logrará captar la información necesaria

para poder valorar y verificar la teoría planteada. La entrevista que se aplicó fue una estructurada, con una guía de preguntas específicas, en el mismo se indica que se pregunta y en qué orden. Para mayor comprensión la Figura 9, nos muestra esta técnica.

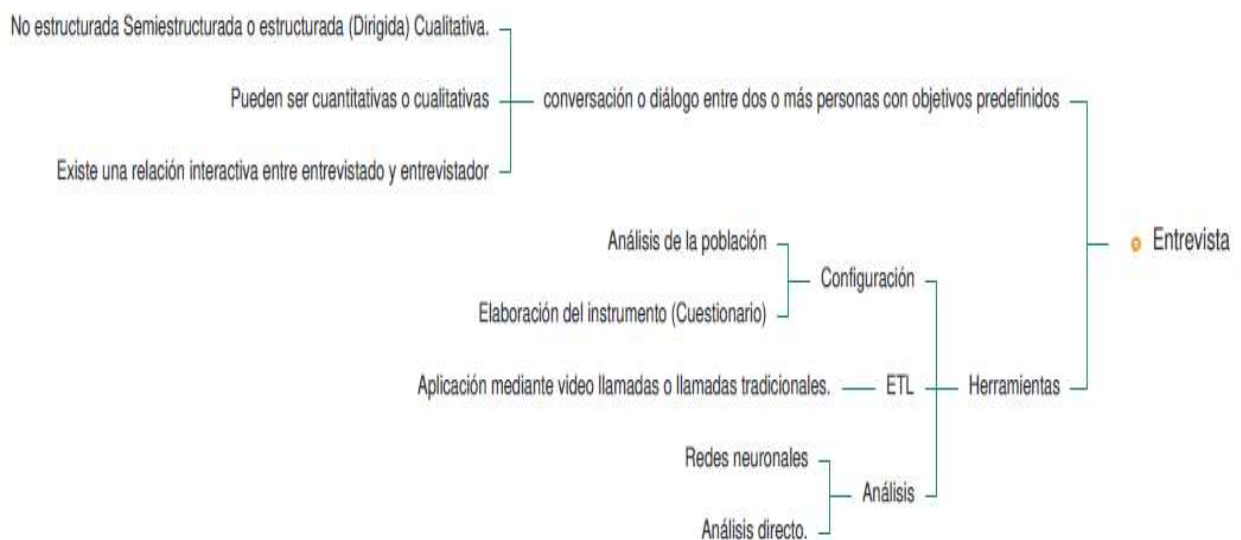


Figura 9. Técnica de recolección de datos, Entrevista, elaboración propia del autor

La guía de la entrevista está dada de la siguiente manera:

GUÍA DE ENTREVISTA SOBRE EL USO DE LA PROPUESTA METODOLÓGICA	
<b>Hora y fecha:</b>	
<b>Lugar (ciudad y sitio específico):</b>	
<b>Entrevistado(a) (nombre/edad/género/puesto actual)</b>	
<b>Introducción</b>	La finalidad es recolectar datos, los mismos que servirán para solventar la creación de una propuesta metodológica. Ustedes han sido elegidos porque pertenecen al departamento de desarrollo de la empresa SOLNUS de la ciudad de Loja, sitio en el cual vamos a aplicar nuestra metodología.

La información receptada será de carácter confidencial, su duración será de 20 minutos aproximadamente.

**Preguntas de la entrevista:**

1. ¿Cree usted necesario desarrollar una propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad para el Departamento de desarrollo?
2. ¿Considera usted que, con la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, se optimice la seguridad en el desarrollo?
3. ¿Cree usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, abarca con todas las actividades de desarrollo de software?
4. ¿Cree usted con la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, se optimice la calidad de la seguridad del software desarrollado?
5. ¿Cree usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad permitiría definir el alcance del proyecto a desarrollar?
6. ¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad se entregaría al cliente un software completamente seguro?
7. ¿Considera usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, cumple con los objetivos definidos por el departamento de desarrollo?
8. ¿Con la implementación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, considera usted que se puedan solucionar las falencias de seguridad en el desarrollo de software presentados al momento?
9. ¿Considera usted que, con la Propuesta metodológica, es necesario aplicarlo como una metodología y estándar a seguir?
10. ¿Considera usted que, con la Propuesta metodológica, sería fácil de ser aplicado por el personal técnico de la empresa SOLNUS de la ciudad de Loja?
11. ¿Considera usted que con la Propuesta metodológica guarda los principios para el desarrollo ágil?
12. ¿Considera usted que con la Propuesta metodológica promueve el trabajo en equipo entre el personal técnico y los usuarios?
13. ¿Considera usted que las técnicas de seguridad de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad son de aprendizaje y aplicación fácil?

## 2.5 Métodos Empíricos con los materiales utilizados

Para esta sección se utiliza, el SLR que ya está especificado en el capítulo anterior en la sección que indica la parte de antecedentes históricos, por tal motivo es de tipo **exploratoria**, determinando que no se han realizado estudios anteriores para tratar de resolver el problema concreto. Es por ello que para el estudio en cuestión se ha tomado como herramienta el estudio de caso, mismo que será aplicado en la empresa de desarrollo de software llamada SOLNUS de la ciudad de Loja, esta a su vez, permitirá obtener la información, desde la empresa que hemos definido desde en el cálculo de la población. Cabe destacar que la información recopilada es de carácter confidencial y la información encontrada, no se puede utilizar para cualquier otro tipo de estudio, esta carta de confidencialidad se encuentra en los anexos.

## 2.6 Técnicas Estadísticas para el Procesamiento de Datos Obtenidos

Al ser una investigación exploratoria, con un diseño cuasiexperimental y en función de que la muestra no es probabilística. El método estadístico para procesar los datos obtenidos con la utilización de la entrevista utilizada para captar estos datos, es el de *Chi Cuadrado* el mismo que nos permitirá hacer la prueba de hipótesis respectiva para validar nuestra teoría.

La prueba de hipótesis se realiza con el método Chi Cuadrado, este método se lo detalla a profundidad en líneas posteriores.

- a) **Hipótesis de la investigación:** Si se diseña la metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas, entonces se minimizará vulnerabilidades a los ataques informáticos.
- b) **Variable Independiente:** Diseño de una metodología de desarrollo ágil de software que agregue la dimensión de la seguridad, mediante la aplicación de técnicas de seguridades informáticas
- c) **Variable Dependiente:** Minimización de vulnerabilidades del software frente ataques informáticos

## CAPÍTULO 3 RESULTADOS OBTENIDOS

En este capítulo se detallan los resultados obtenidos en el estudio realizado en líneas anteriores, la fundamentación del aporte práctico y su elaboración.

### 3.1 Selección de las Metodologías Ágiles para esta Investigación

Dentro de los objetivos planteados, para esta investigación, está el de comparar varias metodologías ágiles, teniendo en cuenta su elección, se garantiza la suficiente información y se fundamentan las características y fases que tienen cada una de ellas. El objetivo es seleccionar varias metodologías de desarrollo ágil de software, en las cuales se realiza una comparativa pertinente, con el fin de poder tomar las mejores características y sus fases para diseñar el ciclo de vida que tiene esta propuesta metodológica. Es importante trabajar con metodologías que tengan documentación pues esto facilita la obtención de la información, además que implementen dentro de sus fases alguna técnica de seguridad.

### 3.2 Ciclo de Vida Propuesto

Para este estudio, luego de haber revisado cada una de las metodologías basado en la tesis de grado de maestría del autor Chimarro Chipantiza [53], técnicas y prácticas de desarrollo ágil de software, se observa que la mayoría tienen similitud en sus fases, por tal motivo se propone las siguientes fases, el mismo que consta de cuatro: Análisis de Requisitos, Diseño, Desarrollo y Pruebas y finalmente Despliegue, en cada una de estas fases se introduce una serie de buenas prácticas y técnicas de seguridad. Tal como se muestra En la Figura 10:

Otro de los trabajos relevantes en los cuales esta propuesta se basó, es en el estudio que realiza Ríos et al (2018) [32], en el cual se hace una comparación de varias metodologías de desarrollo, con el fin de obtener las principales cualidades de cada una de ellas y poder encontrar el factor común de ciclo de vida que coincida entre todas. Bajo estas directrices y lineamientos establecidos el cuadro de comparación resulta de la siguiente manera detallado en la tabla 5:

<b>METODOLOGÍA</b>	<b>DESCRIPCIÓN</b>	<b>FASES CDVS</b>
<b>MSDL (Microsoft security development lifecycle)</b>	Metodología creada por Microsoft, la misma que integra la dimensión de la seguridad en fases del ciclo de vida del software, utilizando algunas técnicas propias de la compañía	<ol style="list-style-type: none"> <li>1. Planificación</li> <li>2. Análisis de requisitos</li> <li>3. Diseño</li> <li>4. Desarrollo</li> <li>5. Pruebas</li> <li>6. Implementación</li> <li>7. Operaciones y mantenimiento</li> </ol>
<b>Crystal Methods</b>	Framework utilizado para elaborar y planear el procedimiento del desarrollo de software	<ol style="list-style-type: none"> <li>1. Puesta en escena</li> <li>2. Revisores</li> <li>3. Monitoreo</li> <li>4. Paralelismo y flujo</li> <li>5. Estrategia</li> <li>6. Técnica de puesta a punto</li> <li>7. Punto de vista del usuario</li> </ol>
<b>DSDM Dynamic Systems Development Method</b>	Marco de trabajo para el desarrollo de software ágil	<ol style="list-style-type: none"> <li>1. Pre proyecto</li> <li>2. Ciclo de vida del proyecto</li> <li>3. Post proyecto</li> </ol>
<b>UML Lenguaje Unificado de Modelado</b>	Lenguaje creado para la modelación visual y la semántica completa en sistemas de software complejos	<ol style="list-style-type: none"> <li>1. Análisis de requerimientos</li> <li>2. Análisis de diseño</li> <li>3. Diseño</li> <li>4. Programación</li> <li>5. Pruebas</li> </ol>
<b>SCRUM</b>	Marco de trabajo que se aplica en cualquier tipo de empresa de desarrollo de software	<ol style="list-style-type: none"> <li>1. Inicio</li> <li>2. Planificación y estimación</li> <li>3. Implementación</li> <li>4. Revisión y retrospectiva</li> <li>5. Lanzamiento</li> </ol>
<b>XP Programación Extrema</b>	Metodología cuya característica principal radica en la aplicación de 4 ejes principales como son el costo, tiempo, calidad y alcance	<ol style="list-style-type: none"> <li>1. Análisis</li> <li>2. Diseño</li> <li>3. Programación</li> <li>4. Pruebas</li> </ol>

*Tabla 5. Tabla de metodologías y sus CDVS, Elaboración propia del autor*

De acuerdo a esta comparación y siguiendo los lineamientos de la propuesta metodológica se propone el siguiente ciclo de vida, que es la recopilación y el factor común que todas estas metodologías ágiles tienen en cada una de sus fases estableciéndose de la siguiente manera:



## CICLO DE VIDA DEL SOFTWARE PROPUESTO CON ÉNFASIS EN LA SEGURIDAD



Figura 10. Ciclo de vida del Software de la Propuesta Metodológica, elaboración propia del autor

### 3.2.1 Explicación de las Fases de la Propuesta Metodológica

#### 3.2.1.1 Fase de Análisis de Requisitos

Aquí se analizan los procedimientos, que admiten los elementos para definir un proyecto de software. En esta fase se especifican las características operacionales. Existen diferentes tipos de requerimientos como: interfaces, usuarios, factores humanos, requerimientos funcionales, documentales, recursos, *aseguramiento de la calidad y la seguridad*, esta última la razón del porque se realiza esta investigación. Estos requisitos deben ser asegurables y alcanzables. La característica principal de estos requerimientos es que los desarrolladores indiquen a los diseñadores como es que el cliente quiere que resulte el producto final sobre todo con respecto a la seguridad.

### **3.2.1.2 Fase de Diseño**

Se identifica la arquitectura del software que admita los requisitos, los funcionales, no funcionales y otras restricciones. Se identifican las soluciones tecnológicas para cada una de las funciones del sistema. Se asignan recursos y materiales para cada una de las funciones. Se definen las guías de diseño que identifiquen los componentes críticos para la seguridad aplicando los privilegios mínimos. Durante el diseño, los desarrolladores guían un modelado de amenazas a un nivel de componentes, detectando los activos que son administrados por el software y las interfaces por las cuales se pueden acceder a esos activos.

### **3.2.1.3 Fase de Desarrollo y Pruebas**

Esta fase está vinculada directamente con la fase anterior, aquí el diseño se convierte en código y el producto debe satisfacer los requisitos del diseño previamente definidos y realizar, si es debido, los ajustes necesarios para que en dicho diseño no existan errores o inconsistencias, se determinan los caminos claros a seguir para concretar lo que será el producto final.

Las pruebas, son una parte importante en el software, el objetivo es garantizar el producto desarrollado. Esta etapa verifica la interacción e integración de los componentes, se verifica que los requisitos hayan sido implementados de forma correcta, se identifica y se asegura que los errores fueron corregidos antes de su entrega final. las pruebas no es una etapa sencilla, aquí se experimentan las funcionalidades de los primeros prototipos, su objetivo radica en descubrir los errores que no se encontraron en las primeras fases del ciclo.

### **3.2.1.4 Fase de Despliegue**

Para algunas metodologías, el proyecto finaliza en la fase que antecede a esta, en otros sin embargo es preciso influir sobre el comportamiento del cliente y de los usuarios del producto para que éstos lo adopten. Los objetivos fundamentales de esta fase son, conseguir que el producto sea implantado, utilizado por los usuarios y asegurar que los beneficios alcanzados gracias al proyecto se mantengan.

### **3.3 Propuesta Metodológica, Ciclo de Vida, Fases que la conforman y Las Técnicas de Seguridad**

En esta sección se puntualiza el diseño de un ciclo de vida del software, resultado de la comparación que se estableció en líneas anteriores referente a las diferentes metodologías y las técnicas o buenas prácticas de seguridad, que se acoplen al desarrollo de software habitual. También se proporciona una guía para la implementación de la metodología propuesta en un caso práctico para su aplicación.

El objetivo consiste en introducir la dimensión de la seguridad del software en cada una de las fases que se proponen producto de la comparación de las metodologías investigadas, de modo que la entrega de los artefactos no se afecte y la seguridad esté vinculada en cada parte del ciclo de vida.

#### **3.3.1 Organización de la Propuesta Metodológica**

La propuesta metodológica entre la ingeniería de requisitos de seguridad y el ciclo de vida propuesto, estará establecido en el ciclo de vida diseñado. Esto significa que, al momento de implementar esta propuesta el equipo de desarrollo, debe tener conocimientos previos sobre las metodologías de desarrollo ágil y conocer argumentos de seguridad, sin embargo, se realizara una capacitación sobre el tema a implementar. Esta capacitación tendrá los siguientes temas: Diseño seguro, Modelos de riesgos, Codificación segura, Pruebas de seguridad y Privacidad. En lo que respecta a las técnicas de seguridad que se agregaran en las fases del ciclo de vida del software, se estableció las mismas por las siguientes razones:

- Fácil uso para el equipo de desarrollo
- La implementación es de forma gratuita
- Software libre
- Se utilizan con más frecuencia en las metodologías comparadas
- Las técnicas son actualizadas con resultados favorables de acuerdo a estudios realizados por varios trabajos de investigación.

Para mayor comprensión se las describe en la Tabla 6:

FASES DEL CVDS	INGENIERÍA DEL SOFTWARE	DESCRIPCIÓN	PRACTICAS / TÉCNICAS DE SEGURIDAD
<b>ANÁLISIS DE REQUISITOS</b>	<ul style="list-style-type: none"> <li>Especificación de requisitos del software</li> </ul>	Identificar los requisitos de seguridad en la fase de levantamiento o en el desarrollo	<ul style="list-style-type: none"> <li>Establecer requisitos de seguridad</li> <li>Guía de Evaluación de los riesgos de seguridad</li> </ul>
<b>DISEÑO</b>	<ul style="list-style-type: none"> <li>Arquitectura</li> <li>Interfaz</li> </ul>	Arquitectura de seguridad en todas las capas:	<ul style="list-style-type: none"> <li>Patrones de Diseño</li> <li>Modelado de amenazas TAM</li> </ul>
<b>DESARROLLO Y PRUEBAS</b>	<ul style="list-style-type: none"> <li>Programación</li> <li>Pruebas</li> </ul>	Utilizar herramientas automatizadas, prestar atención a los falsos positivos, exigir revisiones de códigos de seguridad.	<ul style="list-style-type: none"> <li>Herramientas de código estático</li> <li>Herramientas de ofuscamiento de código</li> <li>Pruebas de penetración</li> <li>Pruebas de carga</li> </ul>
<b>DESPLIEGUE</b>	Soporte técnico para el paso a producción	Crear y desarrollar entornos de desarrollo, pruebas y producción	<ul style="list-style-type: none"> <li>Plan de respuesta a incidentes</li> </ul>

Tabla 6. Introducción de las técnicas de seguridad, elaboración propia del autor

### 3.3.2 Procesos de Trazabilidad de la Propuesta Metodológica

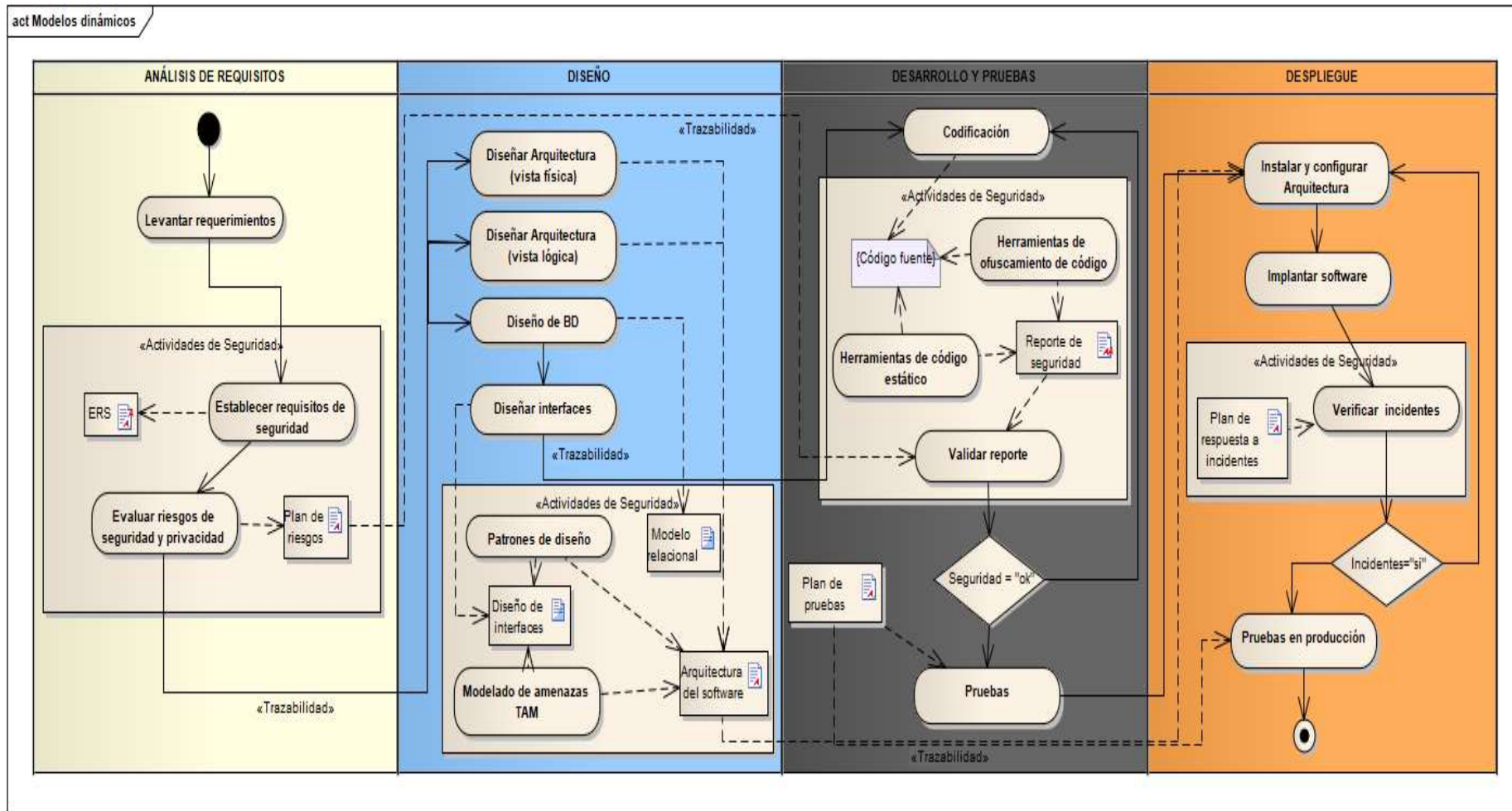


Figura 11. Proceso de Trazabilidad de la Propuesta Metodológica, elaboración propia del autor

### 3.3.2.1 Explicación Del Diagrama De Trazabilidad De La Propuesta Metodológica

El diagrama de trazabilidad, diseñado para esta propuesta metodológica, cuenta de las 4 fases: Análisis de requisitos, Diseño, Desarrollo & Pruebas y finalmente el despliegue. Cada una de estas fases esta entrelazada a su inmediato anterior por sus respectivos entregables, es así que, para poder avanzar en el desarrollo de la metodología, se debe cumplir con el 100% de la fase en la que se encuentra.

#### **Análisis de Requisitos**

Esta fase está compuesta por los procesos y métodos que lleva la ingeniería del software, la particularidad de la propuesta está en la sección de seguridad de cada fase como se muestra en el diagrama principal. Lo primero que debemos hacer como en toda metodología de desarrollo ágil es el levantamiento de los requerimientos, dentro de estos viene las actividades de seguridad propuestas que son el establecer todos los **Requisitos de seguridad**, que el proyecto amerite a estos requisitos que deben ser funcionales y no funcionales, además de tener otras características. Seguido de esto la propuesta sugiere una **Evaluación de riesgos** de seguridad y privacidad a fin de poder determinar desde la primera etapa el nivel de los mismo y sus repercusiones a futuro, el artefacto entregable para esta fase es *Plan de riesgos* el mismo que lleva un control detallado de todas las anomalías encontradas en esta etapa, este plan se lo encuentra en el anexo 1 de esta investigación.

#### **Diseño**

Una vez que hemos evaluado los **Riesgos de seguridad y privacidad** se ha completado la primera fase dando paso a la Diseño, en la cual el desarrollador, de acuerdo a las reglas que establece la ingeniería del software, procede a crear el diseño de la arquitectura del software, esta será de dos tipos: Vista física y lógica, además también se detalla el Diseño de la Base de Datos a utilizar en el proyecto, así como también el Diseño de Interfaces. Sin estos tres diseños ningún proyecto de desarrollo de software esta completo. Dentro de esta fase la propuesta metodológica sugiere las siguientes técnicas para poder introducir la seguridad en el diseño.

La primera técnica que se sugiere es el uso del **Patrón**, un patrón es una solución reusable a un problema recurrente, existen diferentes patrones para cada una de las fases de las metodologías, algunos patrones como adaptador o método plantilla, estos patrones ofrecen soluciones a problemas específicos en el código, como garantizar una única instancia de

un determinado objeto o existen patrones de diseño que permiten cambiar entre distintos algoritmos. En el diagrama que antecede a este texto, observamos que tenemos patrones de diseño para la arquitectura aquí se utilizara el patrón de arquitectura por capas, ya que este nos permite tener una conexión directa con el usuario por medio de la capa de presentación, luego la capa de negocios, la de persistencia y finalmente la de datos.

Para la seguridad completa de esta propuesto se determinó utilizar el **patrón de Diseño Modelo – Vista – Controlador**, este método, permite introducir seguridad en el diseño de las interfaces por medio de sus variantes como son el controlador y el modelo a seguir, existen diferentes formas de diseñar interfaces, pero la más recomendada en la actualidad es el MVC ya que se acopla a cualquier *framework* de programación avanzada. La siguiente técnica para establecer seguridad, es el modelado de amenazas para esto se utiliza la herramienta de análisis de modelado conocida como TAM por sus siglas en inglés, el objetivo de esta técnica es el de ayudar a identificar y planear de forma correcta, la forma más adecuada para mitigar las vulnerabilidades, mediante un enfoque avanzado el mismo que permite obtener varios reportes y sugerencias que son claves para afrontar y eliminar este tipo de anomalías. la consecución y ejecución de estas herramientas nos permite diseñar las interfaces adecuadas con la seguridad exacta y así poder avanzar a la siguiente fase.

### **Desarrollo y Pruebas**

En esta sección el desarrollador realizara su código de forma habitual como lo indica la ingeniería del software, la particularidad que la propuesta metodológica sugiere es el uso de algunas técnicas que permiten darle mayor seguridad a la creación de este código y a su vez, saber si existen anomalías en el mismo. Las técnicas que se usaran para este objetivo son las de **Ofuscamiento**, de **Código estático**, las mismas que mediante la simulación del ambiente permitirán dar pruebas exhaustivas de seguridad al código fuente permitiendo de esta forma que el desarrollador sepa si existe alguna vulnerabilidad por donde se pueda introducir una amenaza. Todas estas actividades nos generan un reporte el mismo que se encuentra detallado en los anexos con el nombre *Reporte de seguridad*, este reporte está vinculado de manera directa con el plan de riesgos que se generó en la etapa de análisis de requisitos.

En esta fase también están incluidas las pruebas al código desarrollado, una vez que el uso de las herramientas ha generado un reporte valido, se procede a ejecutar el plan de pruebas

que también se encuentra en los anexos, este plan de pruebas nos indicara si el código desarrollado está listo para la siguiente etapa de despliegue.

## Despliegue

Una vez terminado y validado el Plan de pruebas se procede a la fase de despliegue, la misma que reúne la información que se vino generando en las fases que la anteceden ya que con el plan de pruebas de la fase de Desarrollo & pruebas, se procede a la ejecución e instalación de un ambiente para el producto final. Las técnicas de seguridad que sugiere la propuesta metodológica en esta sección es la verificación de incidentes integrada, el objetivo es realizar un plan detallado con los incidentes generados en el transcurso del cumplimiento del ciclo de vida del software, este plan se encuentra detallado en los anexos.

Una vez realizado este plan de incidentes, que cabe destacar que está vinculado directamente con el plan de pruebas que generó la fase anterior, el desarrollador puede determinar de forma directa si el producto esta apto para la producción final y puesta en marcha, caso contrario el ciclo de iteración se repite las veces que sean necesarias hasta que el mismo plan de incidentes genere un informe favorable del producto que será entregado al usuario final.

Una vez determinado como se va a efectuar la introducción de las buenas prácticas o técnicas seguras, debemos revisar como quedan estas técnicas, dentro de cada fase del ciclo, como se describe en la Tabla 7:

PRACTICAS / TECNICAS DE SEGURIDAD	CICLO DE VIDA DEL SOFTWARE			
	Análisis de Req.	Diseño	Desarrollo/ Pruebas	Despliegue
a. Establecer requisitos de seguridad	X			
b. Evaluación de los riesgos de seguridad	X			
c. Patrones de diseño		X		
d. Técnica de Modelado de amenazas		X		
e. Herramientas de código estático			X	



f. Herramientas de ofuscamiento de código			X	
g. Pruebas de penetración			X	
h. Pruebas de carga			X	
i. Plan de despliegue				X

Tabla 7. Técnicas de seguridad en las diferentes fases del CDVS, elaboración propia del autor

### 3.3.3 Descripción de las Técnicas Implementadas

#### a) Establecer requisitos de seguridad

La seguridad debe establecerse como una necesidad. Debe estar priorizada, desde el primer instante en la creación de un proyecto de desarrollo, es la parte esencial de un software seguro. Esta etapa es la que define los requisitos, de un proyecto de software. Al definirlos desde una etapa temprana, los miembros que conforman el grupo de desarrollo, podrán determinar los principales resultados e hitos, de modo que los planes en programaciones tengan un mínimo de alteraciones. Al iniciar un propósito se ejecuta el estudio de requisitos de seguridad, el cual contiene las especificaciones mínimas de la aplicación en su entorno operativo, así como también de sistemas de seguimientos tanto a los elementos de trabajo como a las vulnerabilidades de seguridad. La Tabla 8, describe los requisitos que debemos tomar en cuenta el formato es el siguiente.

Historias	Requisito	Detalle de funcionalidad	Categoría	Tipo	Asignada a	Estado

Tabla 8. Plantilla de Análisis de Requisitos de Seguridad, Elaboración propia del autor

**b) Evaluación de los riesgos de seguridad**

Orientada a determinar los sistemas que, en su conjunto o en cualquiera de sus fases puedan estar afectados de manera directa e indirecta por amenazas, valorando los riesgos y estableciendo sus niveles de confianza a partir de las posibles amenazas, las vulnerabilidades existentes y el impacto que pueden causar en el entorno del ciclo de vida del software. Esta técnica consiste en el proceso de comparación del riesgo estimado con los criterios de riesgo y así determinar su importancia. La Tabla 9, describe la plantilla a utilizar para la evaluación de riesgos.

EVALUACIÓN DE RIESGOS									
Nro. Activo	Nombre del Activo	Amenazas	Vulnerabilidades	Impacto	Probabilidad		Controles Implementados	Cálculo de Evaluación	Nivel de Riesgo
				C.I.D	Nivel de amenazas	Nivel de Vulnerabilidades			

*Tabla 9. Plantilla de evaluación de riesgos de seguridad y privacidad, elaboración: propia del autor*

### c) Patrones de diseño

Son soluciones generales repetibles de un inconveniente de la ingeniería de software, están predestinados a asistir al diseño de software menos vulnerable, haciéndolo resistente y tolerante a ataques. Este concepto está relacionado principalmente con la arquitectura, ya que se han desarrollado diseños de patrones para algunas de las fases que comprende el ciclo de vida del mismo, el uso apropiado de esos patrones lleva a la solución de los principales errores de seguridad.

El objetivo de estos patrones es facilitar catálogos de elementos reusables en el bosquejo del sistema de software, evitar reiteraciones en búsquedas a problemas ya conocidos, formalizar los vocabularios que se usan como estándares en el diseño y proporcionar el aprendizaje. Para esta sección la propuesta metodológica sugiere utilizar el patrón de diseño Modelo – Vista – Controlador (MVC), que se detalla en líneas posteriores como se utiliza en la práctica técnica se utiliza la siguiente plantilla que se describe en la Tabla 12:



Figura 12. Patrón de Diseño MVC, elaboración propia del autor

#### **d) Técnica De Seguridad Modelado De Amenazas**

Dentro de las técnicas que la propuesta metodológica sugiere, tenemos la técnica de Modelado de Amenazas, esta técnica permite al usuario tener una óptica general de cómo podemos detectar las vulnerabilidades, a partir de los diagramas de flujo que habitualmente se proponen en el desarrollo de un proyecto de software. La técnica de modelado sugiere los siguientes pasos básicos para detectar amenazas: primero se deben formar un grupo de análisis, luego, desglosar la aplicación identificando componentes precisos, después establecer las amenazas de cada componente y a su vez asignarles un valor a estas amenazas, paso seguido identificar las técnicas que se utilizan para aminorar estos errores localizados.

#### **e) Herramientas de código estático**

El equipo de proyectos realiza el análisis estático. Este análisis admite revisar el código seguro de forma escalable contribuyendo al aseguramiento de las directivas de codificación segura. Por lo general, el análisis de código estático no puede reemplazar una revisión manual de código. El equipo de debe ser conscientes de los pro y contras que tienen las herramientas de análisis estático y estar preparados para utilizar otras herramientas de revisión humana. En esta investigación se utiliza la herramienta Kuiwan [54], la misma que realiza la revisión de código automatizada y análisis de código estático. Abarca la detección de defectos, la gestión de riesgos y la seguridad de las aplicaciones, con funciones mejoradas de ciclo de vida. Certificada por la organización OWASP.

#### **f) Herramientas de ofuscamiento de código**

La facilidad de comprensión, es la característica más valorada en un proyecto software. Los programadores pueden ofuscar deliberadamente el código para ocultar su propósito, su lógica o valores implícitos incrustados en él, principalmente, para evitar la manipulación, disuadir la ingeniería inversa o incluso para crear un desafío recreativo para alguien que lee el código fuente. La herramienta que se utiliza para esta técnica es ProGuard [55], esta herramienta reduce, ofusca y optimiza el código. Es capaz de optimizar el código de bytes , así como detectar y eliminar instrucciones no utilizadas. ProGuard es software.

## g) Pruebas de penetración

Para esta investigación hemos tomado como referencia utilizar la herramienta WebGoat [54], esta herramienta, se integra por un grupo de lecciones que instruyen al usuario sobre los errores comunes de seguridad que suceden en proyectos de desarrollo. Cada lección simula una aplicación vulnerable, para que los desarrolladores, adquieran los elementos necesarios, y consigan ejecutar el ataque y en algunos casos efectuar la defensa para protegerse. La Figura 13, nos muestra la interfaz



Figura 13. Interfaz de la Herramienta WebGoat

## h) Pruebas de carga

En esta parte utilizaremos la herramienta JMeter de Apache, es un plan de pruebas que nos permite generar una jerarquía de componentes en forma de árbol, esta herramienta se la puede utilizar como prueba de carga, ya que permite el análisis y medición del rendimiento de una diversidad de servicios, con énfasis de seguridad.

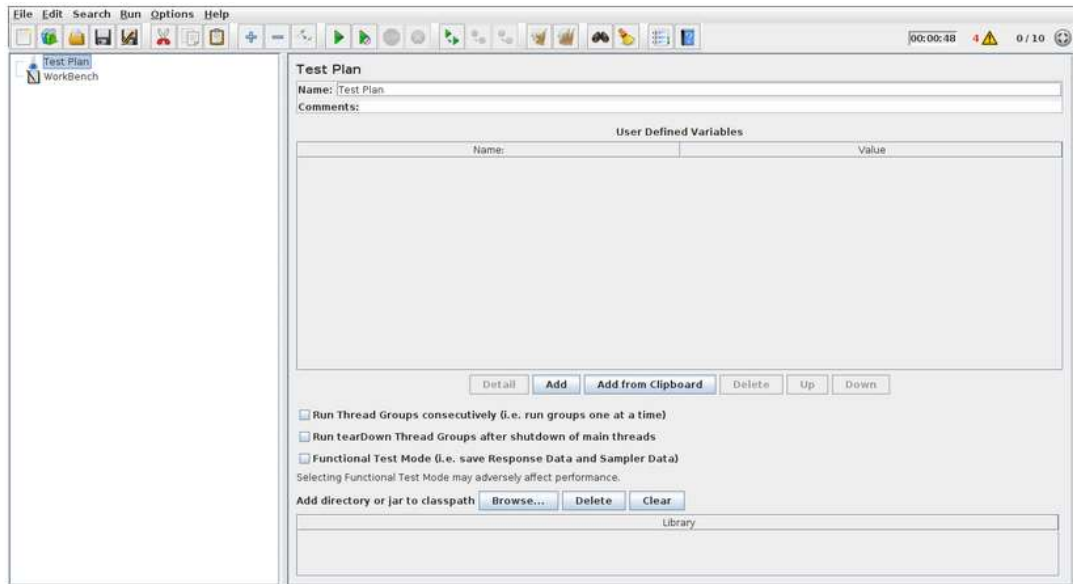


Figura 14. Interfaz del Programa JMeter

## i) Plan de despliegue

Un plan de despliegue representa las acciones asociadas a certificar que el producto de software esté aprovechable para los usuarios, esta fase está vinculada a las demás fases de manera directa en la fase requisitos, ya que esta fase se producen las especificaciones que el usuario quiere del producto final, las pruebas son parte indispensable para el despliegue y los elementos esenciales de las pruebas son parte importante para la entrega de un producto seguro y confiable. Para esta sección se detalla en líneas posterior el plan que la propuesta metodológica sugiere para tener un software seguro y confiable.

### 3.4 Implementación de la Propuesta Metodológica

Para la implementación de la propuesta metodológica, es necesario definir los involucrados que estarán presentes en este proceso. Para esta sección se creó roles y responsabilidades el cual se indica en la siguiente Tabla 10.

Cargo del responsable	Nivel de responsabilidad / Funciones
Personal técnico de la empresa SOLNUS de la ciudad de Loja	Asistir a la capacitación para aplicar la propuesta metodológica
Asesor de la propuesta metodológica (autor de la tesis)	Capacitar al personal asignado
Personal administrativo (Usuarios)	Facilitar la información referente para el progreso del caso particular

Tabla 10. Roles y responsabilidades de la implementación de la propuesta metodológica, elaboración propia del autor

Para seguir un orden estratégico de cumplimiento, se elaboró algunas actividades teniendo como referencia las fases con las que cuenta la propuesta metodológica, es así que estas se describen en la Tabla 11.

Actividad	Tiempo requerido días	Responsable	Técnica / Lugar
Invitación al programa de capacitación sobre los temas que aborda la propuesta metodológica.	1	Tesista de la metodología propuesta	Memorando
Capacitación de la propuesta metodológica	2	<ul style="list-style-type: none"> <li>Tesista de la propuesta metodológica</li> <li>Personal de la empresa SOLNUS</li> </ul>	Instalaciones SOLNUS Loja
Taller de aplicación de la fase de análisis de requisitos	1	<ul style="list-style-type: none"> <li>Tesista de la propuesta metodológica</li> <li>Personal de la empresa SOLNUS</li> </ul>	Instalaciones SOLNUS Loja
Monitoreo y revisión del análisis de requisitos	1	<ul style="list-style-type: none"> <li>Autor de la propuesta metodológica</li> <li>Personal de la empresa SOLNUS</li> </ul>	Validación de la propuesta metodológica
Taller de aplicación de la fase de Diseño	1	<ul style="list-style-type: none"> <li>Autor de la propuesta metodológica</li> <li>Personal de la empresa SOLNUS</li> </ul>	Instalaciones SOLNUS Loja

Monitoreo y revisión del Diseño	1	<ul style="list-style-type: none"> <li>• Autor de la propuesta metodológica</li> </ul> Personal de la empresa SOLNUS	Validación de la propuesta metodológica
Taller de aplicación de la fase de Desarrollo y Pruebas	1	<ul style="list-style-type: none"> <li>• Autor de la propuesta metodológica</li> </ul> Personal de la empresa SOLNUS	Instalaciones SOLNUS Loja
Monitoreo y revisión del Desarrollo y Pruebas	1	<ul style="list-style-type: none"> <li>• Autor de la propuesta metodológica</li> <li>• Personal de la empresa SOLNUS</li> </ul>	Validación de la propuesta metodológica
Taller de aplicación de la fase de Despliegue	1	<ul style="list-style-type: none"> <li>• Autor de la propuesta metodológica</li> <li>• Personal de la empresa SOLNUS</li> </ul>	Instalaciones SOLNUS Loja
Monitoreo y revisión del Despliegue	1	<ul style="list-style-type: none"> <li>• Autor de la propuesta metodológica</li> <li>• Personal de la empresa SOLNUS</li> </ul>	Validación de la propuesta metodológica

*Tabla 11. Plan de Implementación, elaboración propia del autor*

### 3.5 Ejecución de la Implementación

Después de la invitación formal al personal técnico de la empresa SOLNUS de la ciudad de Loja, se efectuó la respectiva capacitación, la misma que fue enfocada a temas referentes a la problemática la cual aborda esta investigación como es la falta del atributo de Seguridad, dentro del ciclo de vida del software. De acuerdo a la ejecución del plan de implementación se efectuó el taller explicativo de las fases con las que conforman la propuesta metodológica y las actividades y técnicas con las que cuenta en cada una de ellas, paso seguido se llevó un monitoreo y revisión, para comprobar que el equipo de desarrollo estuvo aplicando de forma correcta las técnicas propuestas.



### 3.5.1 Estudio de Caso Práctico de la Metodología Propuesta

La teoría de referencia, de la propuesta metodológica se encuentra explicado en las líneas anteriores de esta misma sección, la cual consiste en 4 fases, cada una contiene diferentes técnicas, métodos y buenas prácticas de seguridad. Para el desarrollo de este caso práctico se planteó desarrollar una actividad (4 semanas), con el equipo de desarrollo de la empresa SOLNUS de la ciudad de Loja, conformando un siguiente equipo de desarrollo el mismo que se describe en la siguiente Tabla 12.

N.º	Personal Técnico	Cargo en SOLNUS	Rol en la Propuesta Metodológica
1	Ing. Sist. José Bustillos	Codificador del framework	Integrante del equipo
2	Ing. Sist. Nelson Agurto	Programador master	Integrante del equipo
3	Ing. Sist. Galo López	Analista de redes y telecomunicaciones	Master del equipo (autor de la tesis)
4	Ing. Sist. Michael Méndez	Analista de Planificación	Integrante del equipo
5	Tnlgo. Dis. Kevin Macas	Diseñador Grafico	Integrante del equipo
6	Ing. Sist. María Paz Ludeña	Analista comercial	Testing

Tabla 12. Equipo de desarrollo, elaboración propia del autor

La iteración planteada en el tema a desarrollar, se puso en debate y se eligió una migración de tecnologías de la plataforma.

### 3.5.2 Fase de Análisis de Requisitos de Seguridad

En lo propuesto para esta fase según lo descrito en líneas anteriores de la propuesta metodológica, la técnica de seguridad descrita para esta sección es análisis de requisitos de seguridad, el objetivo de esta técnica es elaborar una lista con los requisitos tanto funcionales como los no funcionales (no dejando el atributo de calidad), las prioridades que tienen estos, las categorías en las que están clasificados, el tipo de requisito si es una mejora o es un nuevo y la persona encargada de identificar este requisito. Para este ejemplo se detalla los requisitos que se tomaron en cuenta en la empresa SOLNUS.

Historias	Requisito	Detalle de funcionalidad	Categoría	Tipo	Asignada a	Estado
3	RQ01	El sistema debe permitir el ingreso solo a usuarios autorizados	* Seguridad	Mejora	José Bustillo	Planificado
2	RQ02	el sistema deberá proporcionar visores adecuados para la lectura de documentos en el almacén de datos	Funcionales	Consulta en pantalla	José Bustillo	En proceso
2	RQ03	en cada periodo de tiempo se deberá asignar un identificador único para que los usuarios tengan niveles de acceso en el sistema	* Seguridad	Mejora	José Bustillo	Culminado
3	RQ04	La interfaz de usuario debe tener fluidez y tiempo de respuesta pertinente.	Interfaces	Consulta en pantalla	Michael Jiménez	Culminado
2	RQ05	El sistema deberá ofrecer el uso sencillo e intuitivo.	Interfaces	Consulta en pantalla	Michael Jiménez	Culminado
3	RQ06	El sistema deberá ser estable y responder a los posibles fallos que se produzcan.	Funcionales	Mejora	Galo López	Culminado
3	RQ07	Disposición del sistema para prestar servicio correctamente.	Funcionales	Mejora	José Bustillo	Culminado
3	RQ08	Continuidad del servicio prestado por el sistema	* Seguridad	Mejora	Michael Jiménez	En proceso
2	RQ09	Ausencia de alteraciones inadecuadas al sistema	Funcionales	Nueva/Mejora	Galo López	Culminado

Tabla 13. Plantilla de Análisis de Requisitos de Seguridad

El siguiente paso a seguir en esta fase de análisis de requisitos, es la evaluación de riesgos de seguridad, el objetivo principal de esta técnica consiste determinar la importancia del riesgo.

Lo primero que tenemos que hacer es una valoración de los activos que tenemos en la actualidad.

VALORACIÓN DE LOS REQUISITOS							
Nro. Requisito	CATEGORIA	Tipo de soporte	Ubicación	Valoración del requisito			
				C: Confidencial			
				I: Integral			
				D: Disponible			
Cf	Int.	Disp.	Valor				
RQ01	Seguridad	Físico y Lógico	Sistema General	2	2	2	2
RQ03	Seguridad	Físico y Lógico	Sistema General	1	2	3	2
RQ08	Seguridad	Físico y Lógico	Sistema General	2	3	2	2

Tabla 14. Valoración de activos de la información, elaboración propia del autor

Luego de esto se establecen la probabilidad de los criterios de ocurrencia de amenazas, en la Tabla 15, se describen estos niveles.

Nivel de amenazas	Criterios de probabilidad	Criterios por ocurrencia (condición)	Criterio por atractivo	Eje
<b>Alto (3)</b>	La ocurrencia es muy probable (probabilidad > 50%)	Bajo circunstancias normales	El atacante se beneficia en gran medida por el ataque	Código malicioso
<b>Medio (2)</b>	La ocurrencia es probable (probabilidad =50%)	Por errores descuidos	El atacante se beneficia de alguna manera por el ataque	Falla de hardware
<b>Bajo (1)</b>	La ocurrencia es menos probable (probabilidad >0 y <50%)	en rara ocasión	El atacante no se beneficia del ataque	causas naturales

Tabla 15. Niveles de amenazas, elaboración propia del autor

El siguiente paso es crear niveles de vulnerabilidad como se describe en la Tabla 16:

Nivel de vulnerabilidad	Criterio	Ejemplo
<b>Alto</b>	No existe ninguna medida de seguridad implementada para prevenir la ocurrencia de la amenaza	No se utilizan contraseñas para que los usuarios ingresen a los sistemas
<b>Medio</b>	Existen medidas de seguridad implementadas que no reducen la probabilidad de ocurrencia de la amenaza a un nivel aceptable	Existen normas para la utilización de contraseñas, pero no se implementa
<b>Bajo</b>	La medida de seguridad es adecuada	Existen normas para la utilización de contraseñas y es aplicada

Tabla 16. Niveles de Vulnerabilidades, elaboración propia del autor

Una vez definidos todos los criterios de vulnerabilidad, de amenazas y los activos procedemos a determinar valor del riesgo, para esto se utiliza la siguiente formula:

Valor de Riesgo = (Valor del requisito) \* (Nivel de amenaza) \* (Nivel de vulnerabilidad)

Valor de Riesgo	
1 a 3	Riesgo Bajo
4 a 7	Riesgo Medio
8 a 10	Riesgo Alto

Tabla 17. Niveles de Riesgos, elaboración propia del autor

Finalmente, la Tabla 18, se refiere la forma como quedan evaluados los riesgos, la misma se utiliza en todos los proyectos:

EVALUACIÓN DE RIESGOS									
Nro. Activo	Nombre del reque.	Amenazas	Vulnerabilidades	Impacto Cf. Integ. Dispon.	Probabilidad		Controles Implementados	Cálculo de Evaluación	Nivel de Riesgo
					Nivel de amenazas	Nivel de Vulnerabilidades			
RQ01	Seguridad, en los ingresos de usuarios	Indisponibilidad de servicios	Red de datos mixta (cat. 5e, 6a)	2	2	2	Mantenimiento local	8	Alto
RQ03	Seguridad, en la asignación de tiempos	Desarrollo de nuevas funcionalidades para la gestión de TH	Incompatibilidad del software base con plataforma de desarrollo actual (php)	2	2	2	Soporte contratado	8	Alto
RQ08	Seguridad, en la continuidad	Acceso no deseado a activos críticos	imposibilidad de actualizar firmware por falta de recursos del equipo	2	2	2	Mantenimiento local	8	Alto

Tabla 18. Tabla General de Evaluación de Riesgos, elaboración propia del autor

### 3.5.3 Fase de Diseño

La propuesta metodológica se plantea utilizar el patrón de diseño, modelo, vista controlador, ya que permite al desarrollador, llevar el código de una manera formal, así como también permite tener una estructura del proyecto entendible, de ahí por qué existen diferentes patrones de diseño que permiten clasificar y separar los módulos para de esta forma llevar el equilibrio entre los componentes que forman un proyecto. Porque optamos por utilizar este patrón de diseño por 3 razones importantes para cualquier proyecto que se quiera desarrollar:

1. proceso rápido de desarrollo, el proceso de desarrollo es rápido y paralelo, ya que al utilizar el patrón una persona puede trabajar en la vista, otra en el modelo y otra en el controlador y así crear la lógica empresarial, básicamente nuestro proceso de desarrollo será beneficiado
2. las modificaciones no afectan a todo el modelo, cualquier cambio no afectara a la arquitectura de la aplicación, porque la parte del modelo no es dependiente de otro componente como la vista
3. el soporte es asíncrono, MVC ayuda al programador a desarrollar permitiendo tener un rendimiento superior al cargar su contenido.

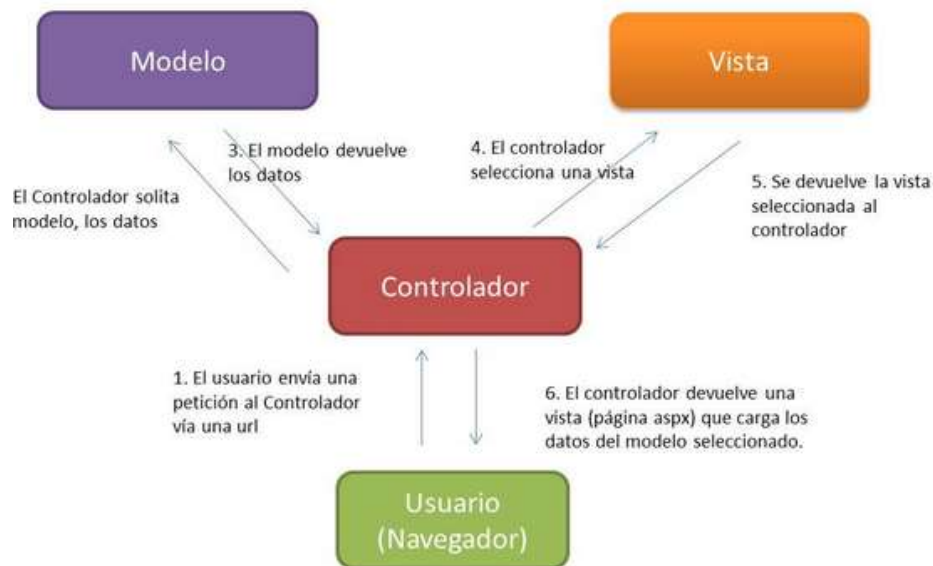


Figura 15. Patrón de diseño Modelo – Vista – Controlador, elaboración propia del autor

## MODELO

Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos, así como las consultas y actualizaciones. También concede los permisos de acceso, toda esta información es enviada a la Vista.

## VISTA

Representa la lógica y la información del negocio, interactúa de forma directa con el usuario también se la conoce como la interfaz de usuario, por lo tanto, requiere que el modelo envíe la información que debe representar al usuario final.

## CONTROLADOR

Responde a sucesos, comúnmente labores que el usuario solicita al modelo, cuando se ejecuta una solicitud sobre la información. También envía los comandos a la vista, es otras palabras, el controlador sirve de intermedio entre la vista y el modelo.

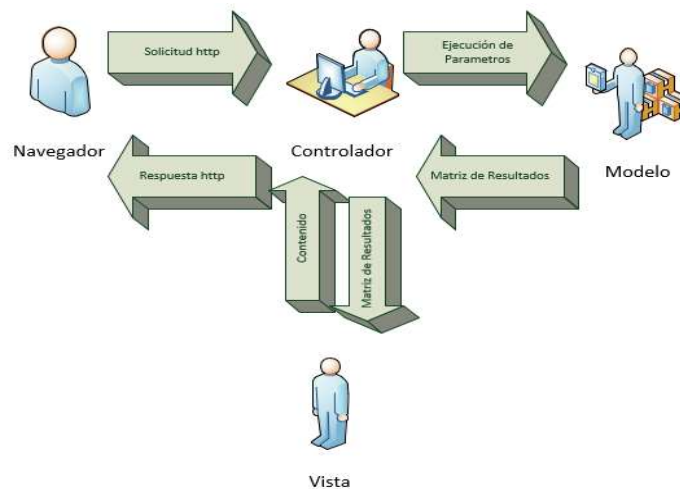


Figura 16. Patrón de diseño implementado, elaboración propia del autor

### 3.5.3.1 Implementación del patrón de diseño MVC

Los pasos que seguiremos para esta implementación son los generales para cualquier tipo de proyecto, en esta ocasión enfocados en la empresa SOLNUS.

**Creación del Modelo;** Primero creamos el modelo que en esta ocasión se denomina LoginUser, y nos permitirá describir las características propias del objeto. Las responsabilidades que tendrá el modelo son:

- La Persistencia de Datos clases que se encuentran presentes en las tablas de base de datos.
- Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si el usuario y contraseña están acreditados en la base de datos se debería consultar el tiempo de respuesta e ingreso al sistema".
- Llevar una bitácora de las vistas y controladores que tiene el sistema.
- Notificar a las vistas los cambios que en los datos pueda producir.

**Crea la vista;** Como se ha descrito anteriormente la vista es también conocida como la interfaz del usuario en este caso creamos la interfaz del aplicativo.



Nunca pares de aprender

Correo electrónico

Contraseña

INICIA SESIÓN

¿Olvidaste tu contraseña?

O también inicia sesión con:

Ingresar con Facebook

Ingresar con Google

¿Nuevo Usuario? **Crear Cuenta**

Al hacer clic en "Crear cuenta certifico que tengo 16 años o más y acepto las Condiciones de Uso, la Política de Privacidad, la Política de Cookies y recibir novedades y promociones."

Figura 17. Interfaz de usuario creada con el patrón MVC

las funciones que realizara son:

- Recoger datos del modelo y los visualiza al usuario.
- Tener una bitácora de todos los controladores asociados.



- Dar servicio de actualización, es decir que el controlador cuando sea llamado por el modelo, cambiara su estado a activo informando todo lo que suceda por errores producidos por agentes externos

**Crear el controlador**, esto nos sirve para tener una ejecución de los eventos correctamente, se crea el controlador y será el intermediario entre el modelo y la vista.

```

478 class NuevaWeb(Categories, FormView):
479     template_name = "nuevaweb/interinas/index.html"
480     authenticate_form = AuthenticationForm
481     createUser_form = UserCreationForm
482     success_url = reverse_lazy("pagweb:mis_cursos")
483     next_page = reverse_lazy("pagweb:nueva_web")
484
485     def get(self, request, *args, **kwargs):
486         lista_grupos = Categories.groupsCategories(self)
487
488         context = {
489             "grupos": lista_grupos,
490             # "authenticate_form": self.authenticate_form,
491             # "createUser_form": self.createUser_form,
492         }
493         return self.render_to_response(context)
494
495     estado = False
496
497     def post(self, request, *args, **kwargs):
498         form1 = self.authenticate_form(request=request, data=request.POST)
499         form2 = self.createUser_form(data=request.POST)
500         if form1.is_valid():
501             login(self.request, form1.get_user())
502             return HttpResponseRedirect(self.get_success_url())
503         elif form2.is_valid():
504             form2.save()

```

Figura 18. Controlador creado con el patrón MVC

Las responsabilidades que tendrá son:

- Recoger los sucesos de entrada como recibir el evento cuando se hace un clic.
- Aquí, se concentran las reglas establecidas en la gestión de eventos. Estas acciones se las toma como peticiones al modelo o a las vistas, como puede ser el proceso de actualización.

**Crear clases principales**, se necesita crear una clase principal para utilizar los métodos del controlador y demostrar el uso del patrón de diseño MVC

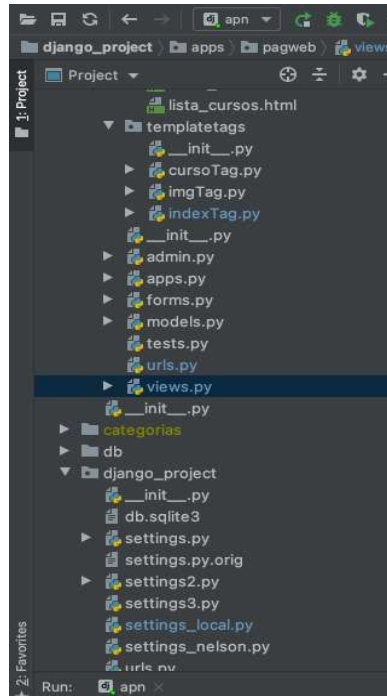


Figura 19. Clases creadas con el patrón MVC

1. Verificar los resultados, en esta parte se muestra que el ejemplo funcione correctamente y como debe ser el resultado correcto. Al final se muestra la pantalla del sistema.

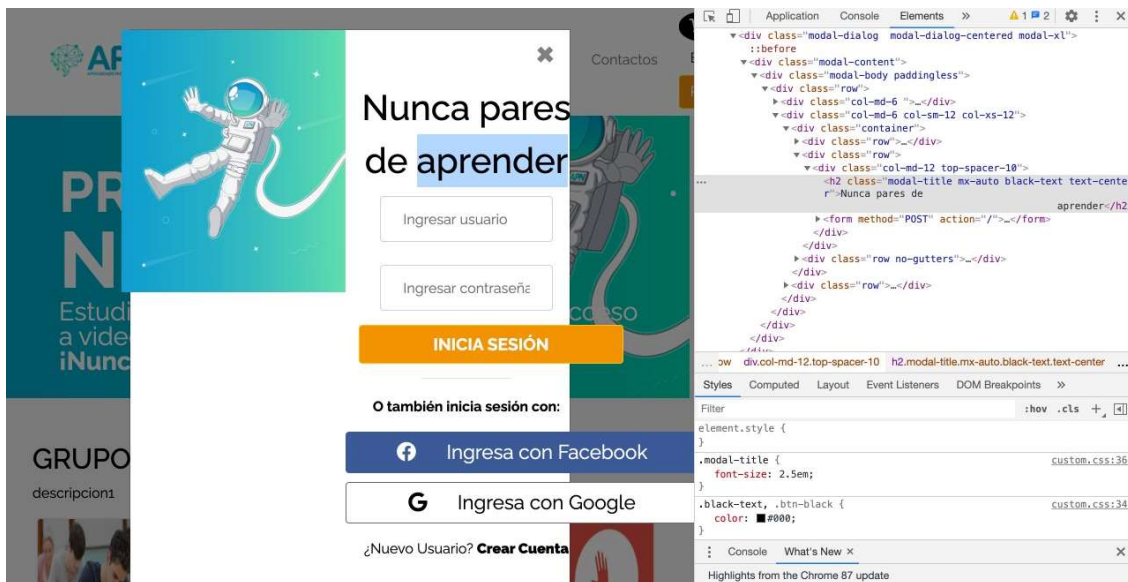


Figura 20. Pantallas finales con el código creado por el patrón de diseño

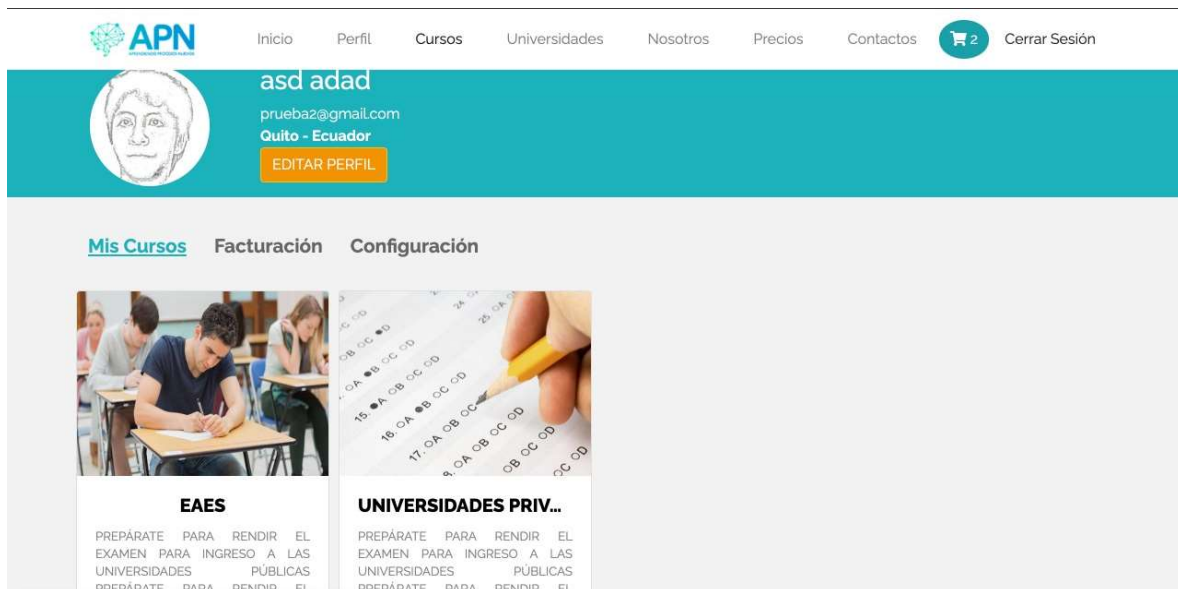


Figura 21. Plantilla de Patrón de Diseño implementada

### 3.5.3.2 Técnica De Seguridad Modelado De Amenazas TAM

La herramienta a utilizar es la Microsoft Análisis y Modelización de Amenazas o conocida por sus siglas en inglés como TAM, lo primero que se debe hacer es definir los componentes particulares, así como la iteración que tienen entre ellos. De esta manera se identifican las potenciales amenazas y a partir de estas se define la estrategia de seguridad práctica que se aplicará para minimizar esta vulnerabilidad. De esta forma los desarrolladores podrán generar modelos establecidos en el código ya conocido, también permite generar reportes a manera de artefactos de seguridad que se tomarán en cuenta en el plan de incidentes.

#### PASOS QUE EJECUTA LA TÉCNICA MODELADO DE AMENAZAS TAM

Primero se debe recopilar la información, se identifican casos de uso como escenarios posibles. Se debe crear diagramas de flujo, a raíz de los casos de uso se crean estos diagramas para asegurar la profundidad de la amenaza.

Segundo creación y análisis del modelo de amenazas, se deben crear grupos conformados por un integrante de cada sección que conforman el equipo de trabajo, el objetivo de este grupo es buscar las amenazas posibles aun no resueltas.

Tercero análisis de amenazas, después de haber analizado las amenazas en los entornos posibles, se procede a darles un valor a cada una de ellas, se puede utilizar categorías para distinguir las amenazas dependiendo del factor en el que se desenvuelvan, para el ejemplo se categorizan de esta manera:

CATEGORIA	DESCRIPCIÓN
<b>Categoría A</b>	Años potenciales
<b>Categoría B</b>	Capacidad de reproducción
<b>Categoría C</b>	Usuarios afectados
<b>Categoría D</b>	Capacidad de descubrimiento

*Tabla 19. Tabla General de Evaluación de Riesgos, elaboración propia del autor*

La pantalla de interfaz que la herramienta presenta es la siguiente en la cual se muestran las categorías y se explica en que consiste cada una, a continuación, se describen cada uno de ellos.

**Sección A**, se encuentra integrada por los roles que manejaran los integrantes del equipo de diseño, sean estos clientes, administradores que son los usuarios internos y los anónimos que son los usuarios externos que también forman parte del equipo.

**Sección B**, se encuentran por los almacenes de datos de la empresa donde se efectuó el desarrollo del modelado

**Sección C**, están todos los componentes que permiten integrar los datos con el sistema

**Sección D**, se encuentran las dependencias externas y los casos de uso

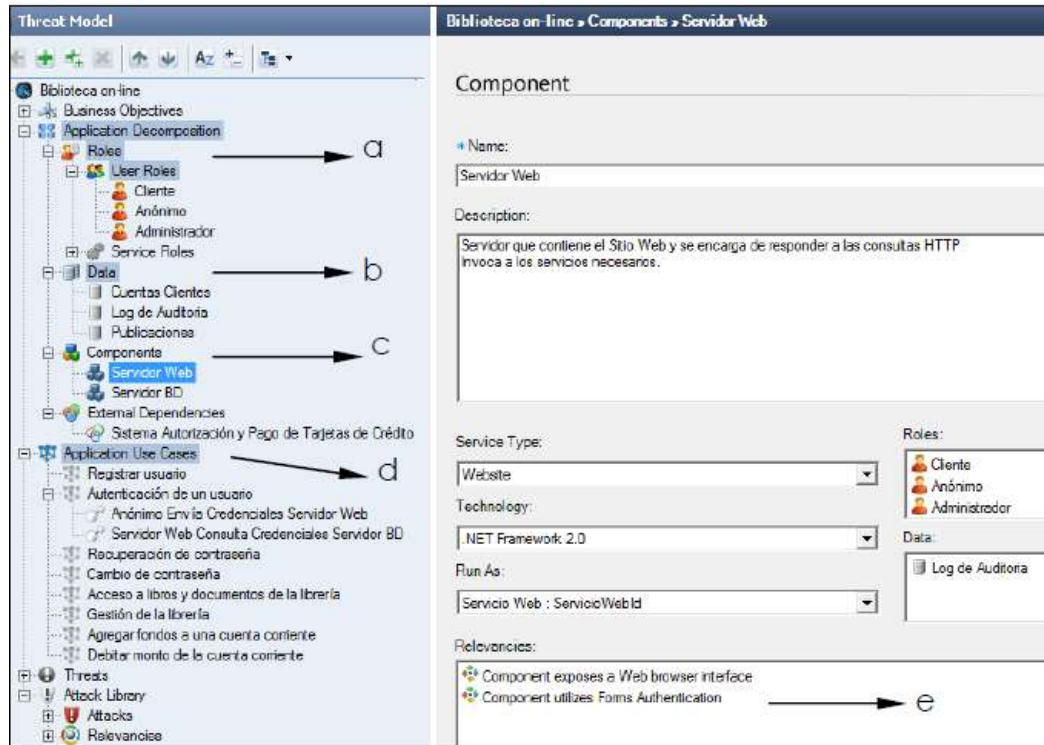


Figura 22. pantalla Principal de la Herramienta TAM, secciones de trabajo

Para el ejemplo práctico haremos resolución del caso de uso de la autenticar un usuario, empezamos creando el diagrama de caso de uso, para este caso la definición de la amenaza será, “revelación de credencial no autorizada”, categorizada como de confidencialidad, luego se procede a determinar el nivel de riesgo que tiene esta amenaza, después se programaran las medidas para mitigar esta amenaza.

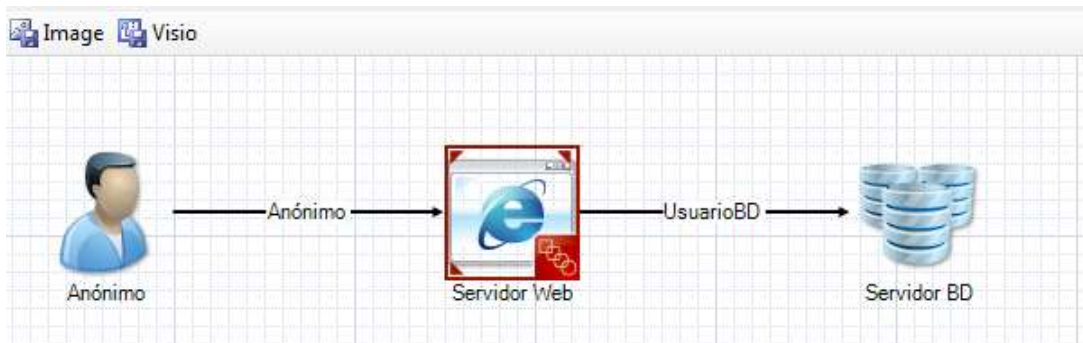


Figura 23. Caso de Uso Autenticar usuario

La herramienta interiormente utiliza el método conocido como DREAD, el mismo que evalúa cinco aspectos importantes, indicados en la tabla 20:

D	Daño potencial
R	Reproducibilidad
E	Explosión (facilidad de multiplicarse)
A	Afectación (usuarios afectados)
D	Descubrimiento (facilidad de revelar)

Tabla 20. Análisis DREAD, elaboración propia del autor

Luego de hacer esta evaluación interna, la herramienta, obtiene el valor de riesgo como un promedio que arroja la amenaza, la imagen siguiente muestra este proceso una vez que ya se ha solicitado el cálculo del valor de riesgo.

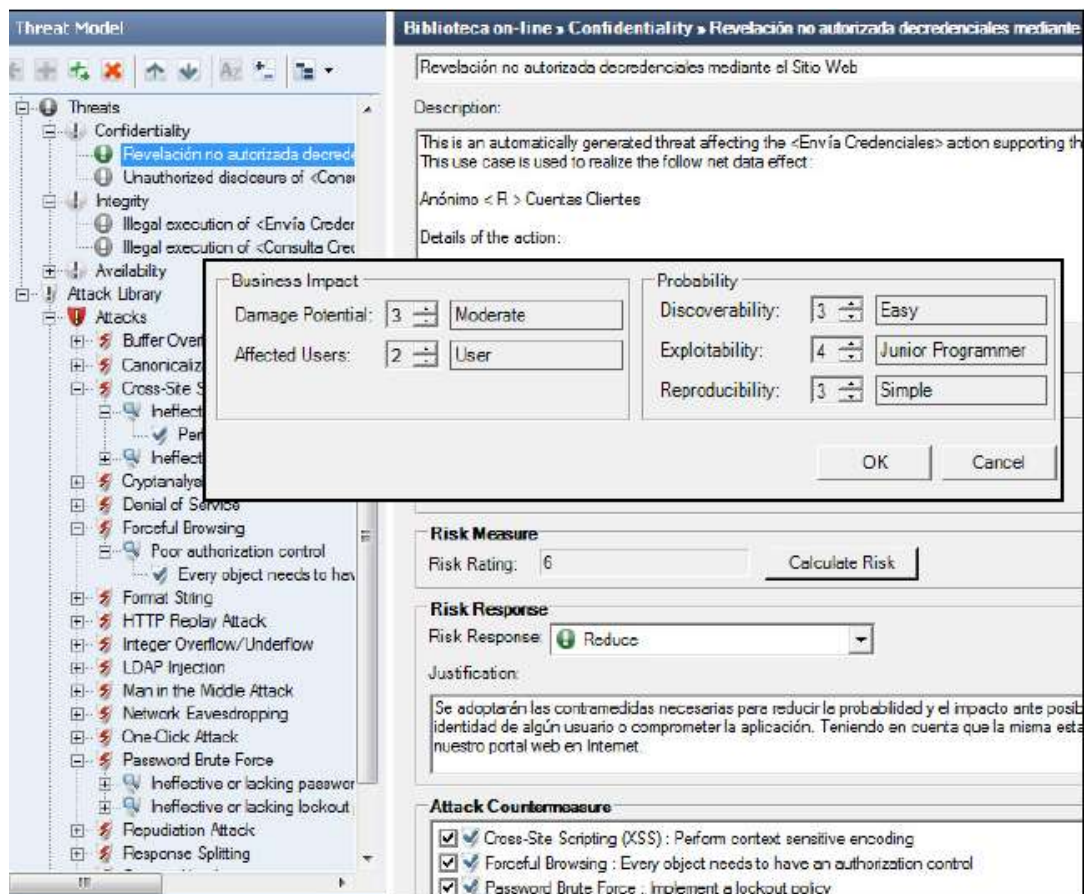
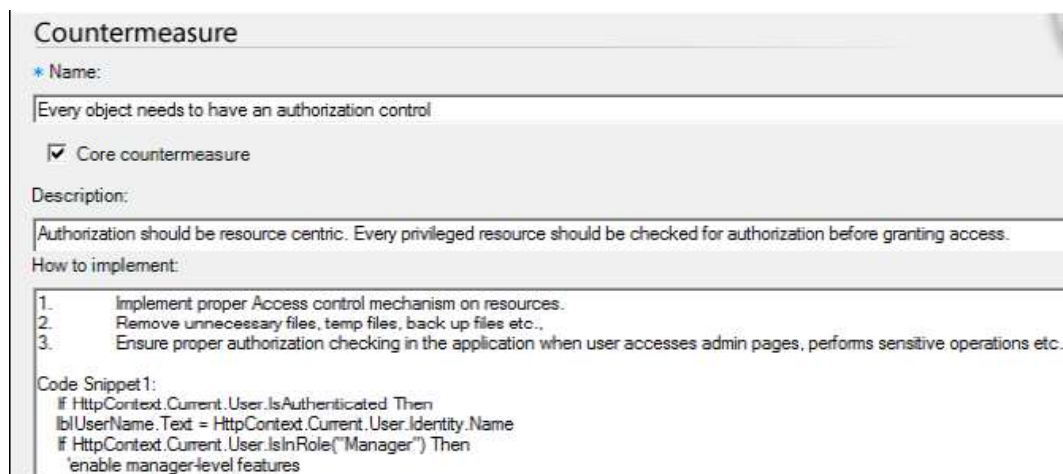


Figura 24. Pantalla de cálculo de vulnerabilidades

Las medidas para mitigar esta amenaza las muestra en la siguiente pantalla, cabe recalcar que se pueden agregar manualmente más contramedidas para afianzar la seguridad. Los informes que la herramienta genera pueden ser generales o en caso particular a algún integrante del equipo



**Countermeasure**

\* Name:  
Every object needs to have an authorization control

Core countermeasure

Description:  
Authorization should be resource centric. Every privileged resource should be checked for authorization before granting access.

How to implement:

1. Implement proper Access control mechanism on resources.
2. Remove unnecessary files, temp files, back up files etc..
3. Ensure proper authorization checking in the application when user accesses admin pages, performs sensitive operations etc.

Code Snippet 1:  
If HttpContext.Current.User.IsAuthenticated Then  
lblUserName.Text = HttpContext.Current.User.Identity.Name  
If HttpContext.Current.User.IsInRole("Manager") Then  
enable manager-level features

Figura 25. Informe de como mitigar la vulnerabilidad

### 3.5.3 Fase de Desarrollo y Pruebas

En esta fase la propuesta metodológica para lo que son las pruebas de código estático sugiere, utilizar la técnica de código estático Kiuwan. Esta plataforma integral de *Software Analytics* para análisis de código estático y revisión de código automatizada admite la exploración del código de seguridad de forma escalable, además asiste a la observación segura de las directivas de codificación. Para esto procedemos a descargar el paquete de instalación de la herramienta. Instalamos el paquete de Kiuwan en nuestro equipo realizamos las configuraciones como se visualiza en la Figura 26:



Figura 26. Pantalla de acceso del Programa Kiuwan

Una vez instalado la herramienta procedemos a ejecutar y analizar nuestro código como se visualiza en la siguiente Figura 27.



Figura 27. Pantalla de análisis de código del programa Kiuwan

Una vez terminado el análisis nos muestra una pantalla donde se describe el análisis completo de la revisión del código, con el número de vulnerabilidades encontradas, como se muestra en la Figura 28:



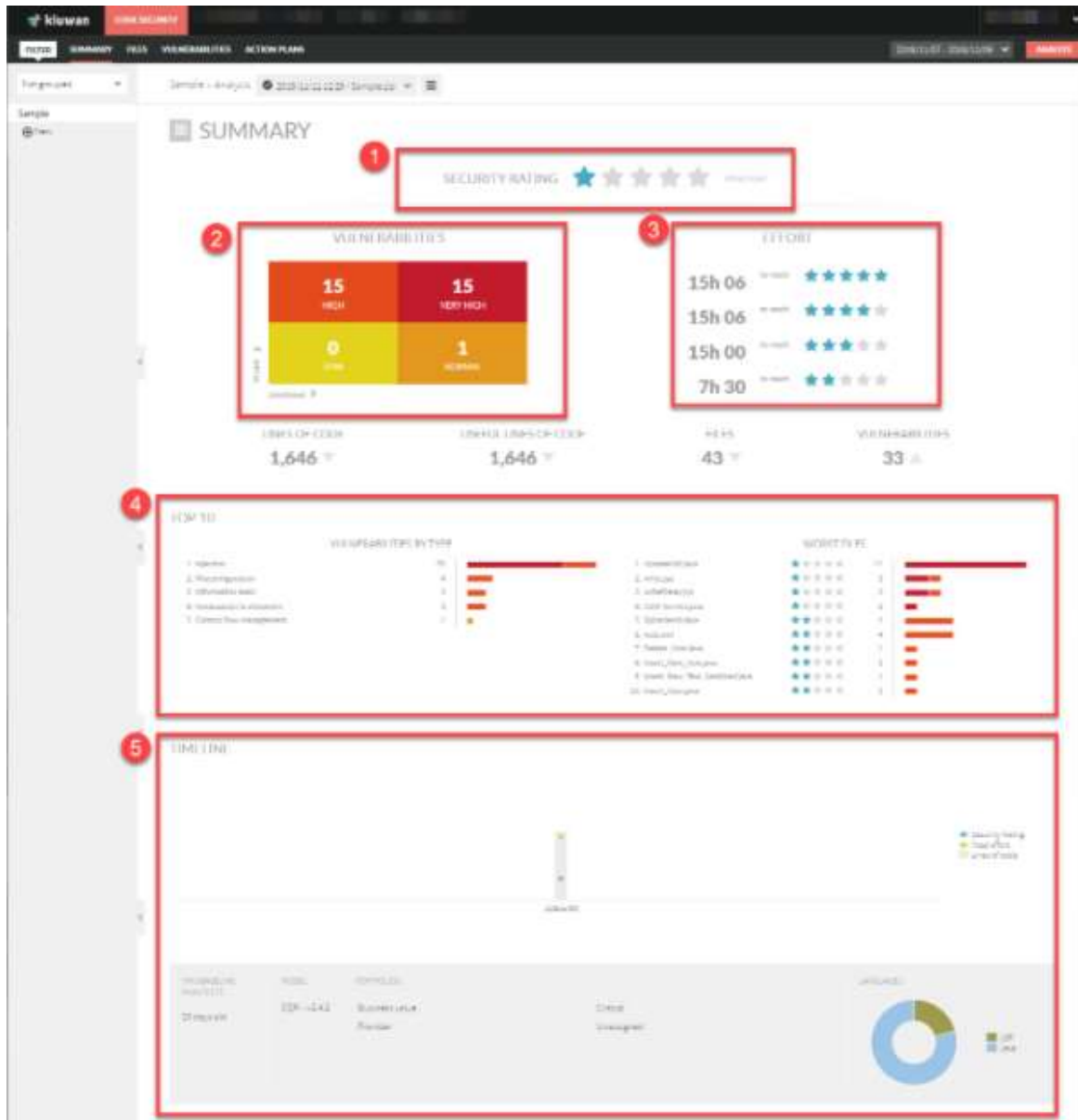


Figura 28. Sumario de análisis del Programa Kiuwan

Como se muestra en la imagen, el sumario comprende varias secciones que la propuesta metodológica explica a detalle para mayor comprensión al momento de ejecutar el desarrollo de cualquier proyecto.

1. Calificación de seguridad; es una calificación discreta que indica al desarrollador, la seguridad que tiene su aplicación en términos de probabilidad e impacto ante las vulnerabilidades, mientras más estrellas tenga mejor seguridad tendrá la aplicación.

2. Vulnerabilidades de seguridad; la herramienta nos muestra un cuadrante en donde se agrupan las vulnerabilidades por probabilidad e impacto, como un plano cartesiano tenemos los dos ejes y la agrupación de los cuadrantes según el nivel obtenido, muy alto, alto, bajo y normal.
3. Esfuerzo; esto se mide en base a los resultados del análisis, este esfuerzo nos indica la inversión de tiempo que se debe emplear para alcanzar los niveles de excelencia para corregir cada vulnerabilidad.
4. Clasificación de vulnerabilidades; en esta sección del sumario nos muestra la clasificación de las principales vulnerabilidades y de los peores archivos. Esta herramienta considera una vulnerabilidad cualquier defecto que pueda producir un problema potencial.
5. Línea de tiempo; en esta parte muestra la evolución de la calificación de seguridad y el esfuerzo realizado con el afán de lograr una mejor puntuación de seguridad.

Para las **pruebas de ofuscamiento**, la propuesta metodológica sugiere utilizar la herramienta *ProGuard*, el motivo es práctico ya que eleva los niveles de seguridad, proporcionando protección a la ingeniería inversa ofuscando los nombres de las clases, campos y métodos que se hayan desarrollado en cualquier tipo de proyecto. Además, ayuda a reducir el tamaño de cualquier aplicación eliminando código no utilizado, se recomienda el uso de esta herramienta no solo en desarrollo sino también en producción.

Para el ofuscamiento del código se debe utilizar algunas librerías que ya vienen instaladas en el paquete original. Proguard cambia el nombre de las clases, campos y métodos por nombres sin sentido lo que dificulta al intruso en el momento que quiera obtener el código deseado.



Figura 29. Pantalla principal de la herramienta Proguard

Implementado quedaría de la siguiente forma como lo muestra la figura 30:

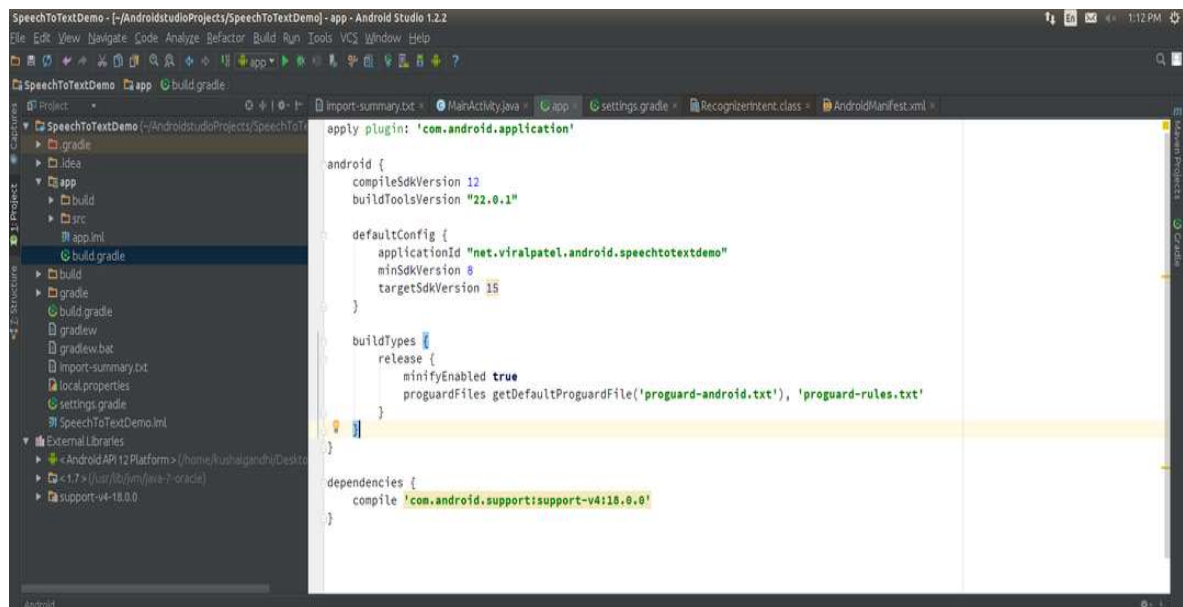


Figura 30. Código principal de ofuscamiento métodos, clases y campo

Como siguiente herramienta para las pruebas y testing que la propuesta metodológica sugiere, tenemos las **pruebas de penetración**, para ello se utilizó la siguiente herramienta dada por la organización OWASP, que son buenas prácticas seguras, como se visualiza en la Figura 31, Webgoat aplicado en la empresa SOLNUS



Figura 31. Interfaz del Programa Webgoat

WebGoat, es una plataforma, diseñada y elaborada por OWASP, para dar lecciones de seguridad a aplicaciones web, cada una de las lecciones que tiene la misma enseña a los usuarios los problemas de seguridad al explotar las vulnerabilidades. En esta sección haremos varias pruebas de vulnerabilidades a la página este mismo proceso se debe seguir en cualquier proyecto de software, y a la vez aporta en elevar el nivel de seguridad deseado.

Iniciamos haciendo un ataque de fuerza bruta, este ataque consiste en ingresar combinaciones de credenciales. Para esto normalmente se utiliza un software personalizado o una herramienta automática que integra palabras de forma sucesiva hasta encontrar una combinación válida, de esta forma se produce un robo de credenciales.

*Proxy – Intercept – intercept is off:*



Figura 32. Interfaz del Programa Burpsuite

Cargar la página para acceder:

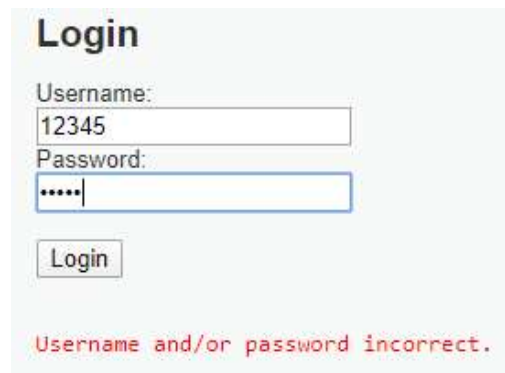


Figura 33. Interfaz del Login del sistema

## Intercept



Figura 34. Ataque de Intercept

Encontramos las cargas útiles para conseguir el intruso en la pagina

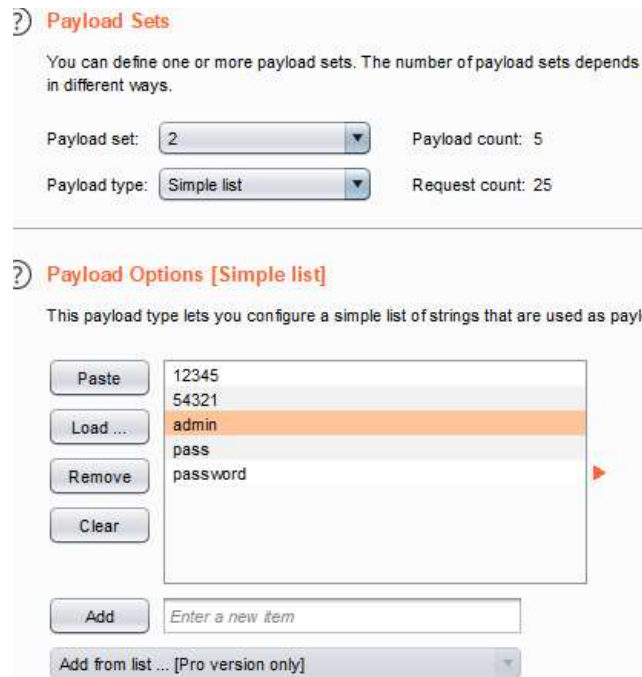


Figura 35. Cargas útiles que muestra el programa

Se realiza correctamente el ataque por fuerza bruta, nos muestra la lista de combinaciones de credenciales encontradas con el ataque realizado al sistema

Intruder attack 2

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4666
1	11111	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
2	22222	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
3	33333	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
4	admin	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
5	administrador	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
6	11111	54321	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
7	22222	54321	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
8	33333	54321	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
9	admin	54321	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
10	administrador	54321	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
11	11111	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4666
12	22222	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4666

Figura 36. Combinaciones de credenciales validas en el ataque

La solución y explicación de este ataque se describe en la Tabla 21:

EXPLICACIÓN DE LA VULNERABILIDAD	
<b>Explicación:</b>	Se trata de aprovechar la vulnerabilidad del ataque a través de la detección de la ip del intruso, luego modificar los parámetros de inicio de sesión con usuarios y contraseña de una biblioteca o repositorio que nosotros creamos para poder aplicar nuestro ataque.
<b>Solución:</b>	<ul style="list-style-type: none"> <li>• Deja de usar el nombre de usuario que viene por defecto "admin", Utiliza una contraseña segura.</li> <li>• realizar regularmente copias de seguridad de archivos y base de datos.</li> <li>• Usa autenticación de dos factores. (Google Authenticator) Protege con una contraseña el WP-Admin y limita los intentos de conexión.</li> </ul>

*Tabla 21. Explicación y Solución a la Vulnerabilidad, elaboración propia del autor*

Otro de los ataques que la propuesta metodológica sugiere son los conocidos SQL Injection, este ataque tiene como objetivo es escudriñar la seguridad del software de la aplicación y permitimos así poder falsear identidades, manipular datos existentes, deshabilitar datos, convertirse en managers de base de datos y poder vulgarizar los datos de cualquier proyecto de software, la Figura 37 muestra la intromisión de código conocido como SQL Injection.

El código a introducir es: `apn.ec/cuenta_usuario/?id menú=22`

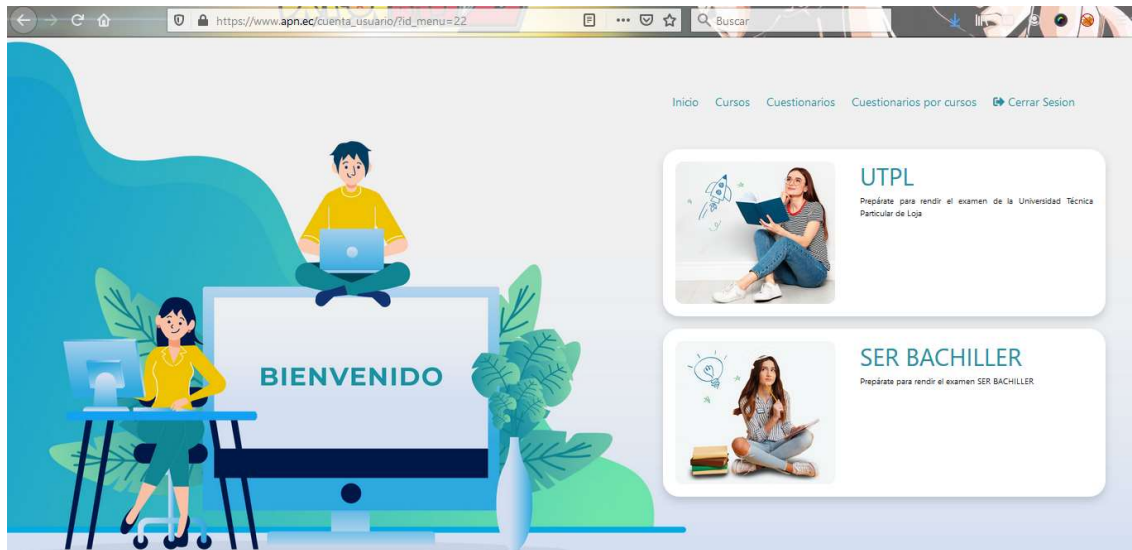


Figura 37. Página Principal del portal APN.EC

La solución a este problema se lo describe en la Tabla 22:

EXPLICACIÓN DE LA VULNERABILIDAD SQL INJECTION	
<b>Explicación:</b>	Este tipo de ataques se explora las vulnerabilidades de la seguridad del software y permitimos así tomar el control absoluto, manejar los datos existentes, convertirnos en managers de la base de datos y poder divulgar completamente la información.
<b>Solución</b>	<ul style="list-style-type: none"> <li>• Filtrar entradas. - Filtrar y comprobar caracteres ingresados por el usuario.</li> <li>• A nivel de código. - se pueden realizar validaciones de los campos y comandos que protejan a través de nuestro código fuente.</li> <li>• Asignar privilegios. - Mínimos privilegios al usuario que se conectara a la base de datos.</li> </ul>

Tabla 22. Explicación y Solución a la Vulnerabilidad SQL INJECTION, elaboración propia

Otro de los ataques que la propuesta metodológica sugiere es el *XSS Reflected*, este ataque Consiste en cambiar los valores que el software usa para pasar a través de otro programa reflejo. Para este caso práctico tenemos la página siguiente.



Insertamos un javascript algo simple como una alerta para ver si nos permite realizarlo recuerde que estamos en nivel bajo y este proceso lo hacemos en la barra de direcciones

?name=<script> alerta ("¡Has recibido XSSed!") </script>



Figura 38. Ejecución del XSS Reflected

La solución de este ataque se lo describe en la tabla 23:

EXPLICACIÓN DE LA VULNERABILIDAD XSS REFLECTED	
<b>Explicación:</b>	Este tipo de vulnerabilidad compromete la seguridad del usuario y no la del servidor. Inyecta código malicioso en el software, con el fin de que el usuario ejecute ese código inyectado en el momento de ver el software reflejada alterando así su forma final.
<b>Solución</b>	Se debe recopilar los datos de requerimientos, no confiables en los campos de salida solo así se puede proteger los cuerpos y atributos, esto permite resolver el problema de esta vulnerabilidad.

Tabla 23. Explicación y Solución a la Vulnerabilidad XSS REFLECTED, elaboración propia del autor

Para las **pruebas de carga** utilizaremos la herramienta *Jmeter*, o también conocidas como pruebas de carga y stress sirven para simular tráfico falso al servidor, de manera que se pueda demostrar cuántos usuarios de manera simultánea pueden estar conectados a la vez sin que exista pérdida de paquetes.

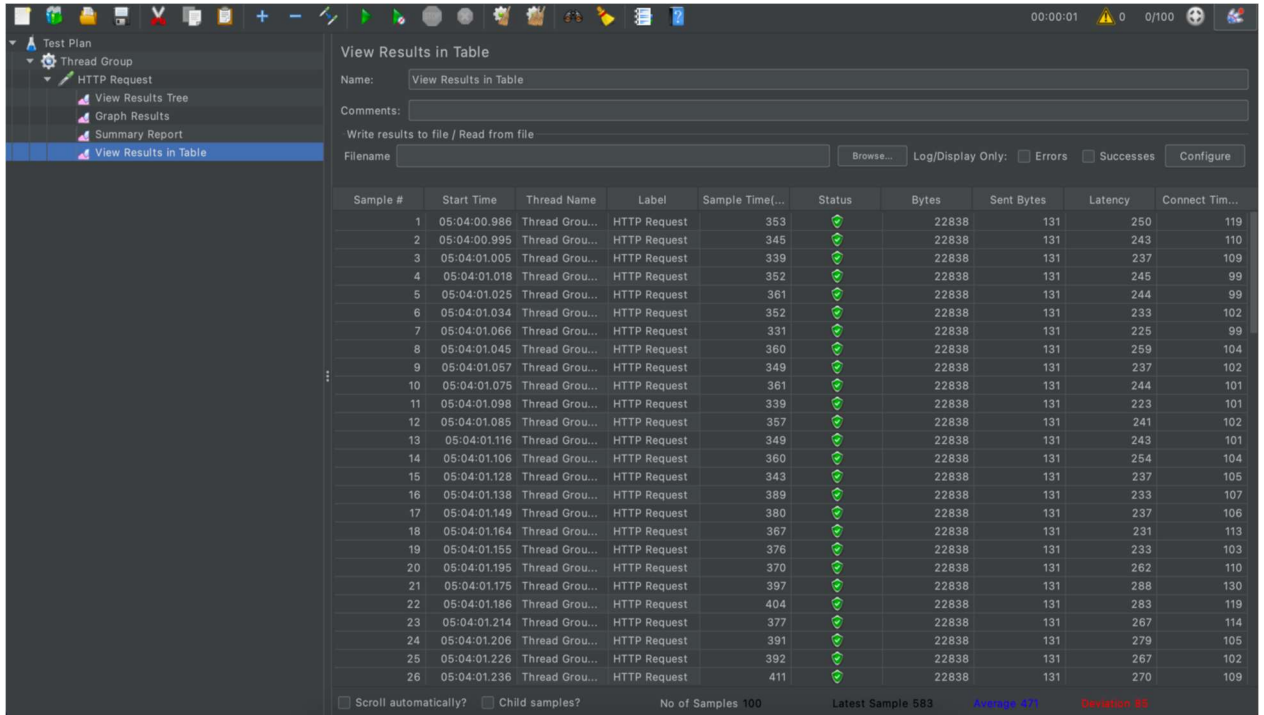


Figura 39. Ejecución del Programa JMeter

Las características del servidor se muestran a continuación, sobre el cual se van a realizar las respectivas pruebas.

Características del servidor.

Memoria RAM	CPU's	SSD DISK	Transferencia
1GB	1	25GB	1TB

Tabla 24. Tabla de características del servidor, elaboración propia del autor

Para iniciar con las pruebas de estrés la aplicación permite asignar ciertos atributos:

- El número de hilos: Usuarios a conectarse
- El periodo: el tiempo en que se enviarán todas las solicitudes, en las pruebas se enviarán  $N$ , cantidad de solicitudes por segundo

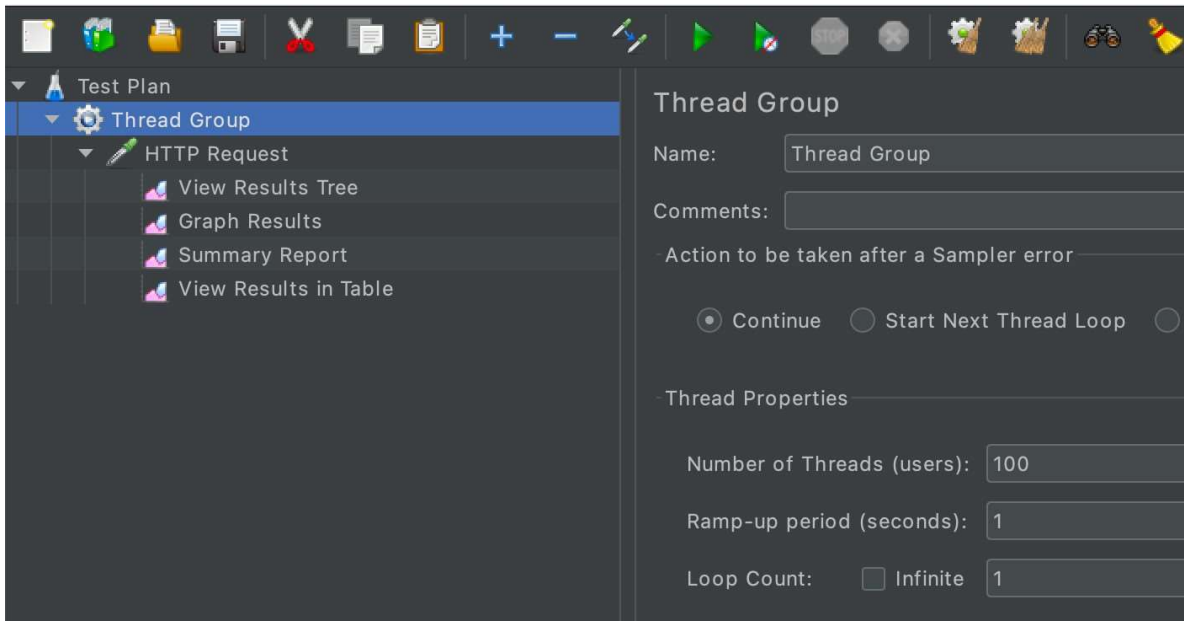


Figura 40. Ejecución de atributos

Luego se añade la petición Http y se ejecuta, se asigna el nombre de dominio, puerto, protocolo y método que se ejecuta, para nuestras pruebas.

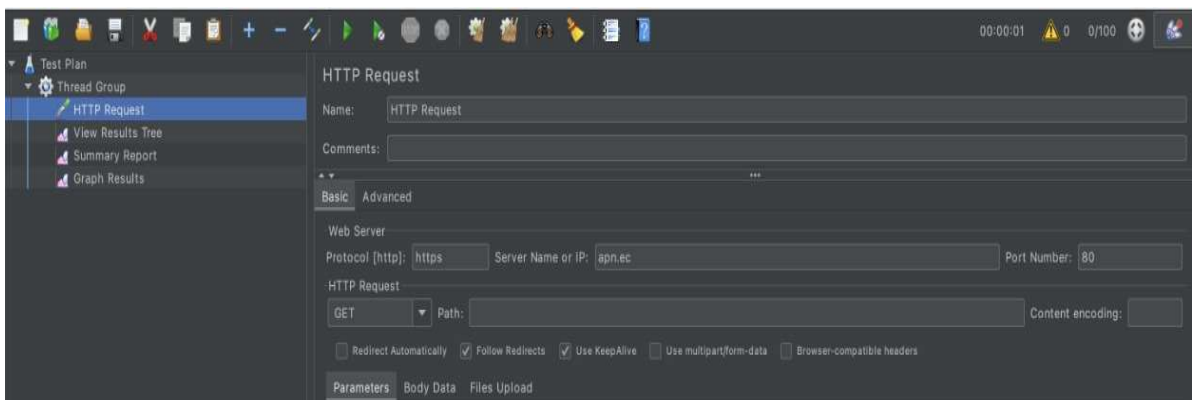


Figura 41. Imagen de las peticiones en JMeter

Una vez ya configurada la herramienta, nos ofrecerá un reporte con el estado de las conexiones se procede hacer las pruebas en intervalos de 50, 100, 150, 200, 500, 1000 y 3000 solicitudes al servidor, de manera que se pueda observar el número de transacciones exitosas y tiempo estimado de ejecución.

Nro. Solicitudes	Nro. Transacciones exitosas	% de Transacciones exitosas	Nro. Transacciones fallidas	% de transacciones fallidas
50	50	100%	0	0%
100	100	100%	0	0%
150	150	100%	0	0%
200	147	73,5%	53	26,5%
500	235	47%	265	53%
1000	457	47,5%	543	54,3%
3000	719	27,3%	2181	72,50%

Tabla 25. Tabla de Transacciones exitosas y fallidas, elaboración propia del autor

Se demostró una tasa de concurrencia alta a través de JMeter Apache y se permite el uso simultáneo de la aplicación hasta de 150 usuarios, con tiempo de respuesta de solicitudes a la aplicación es de 0.8 hasta 18 segundos, si el número de conexiones excede este límite implica una pérdida de solicitudes mayor al 26%.

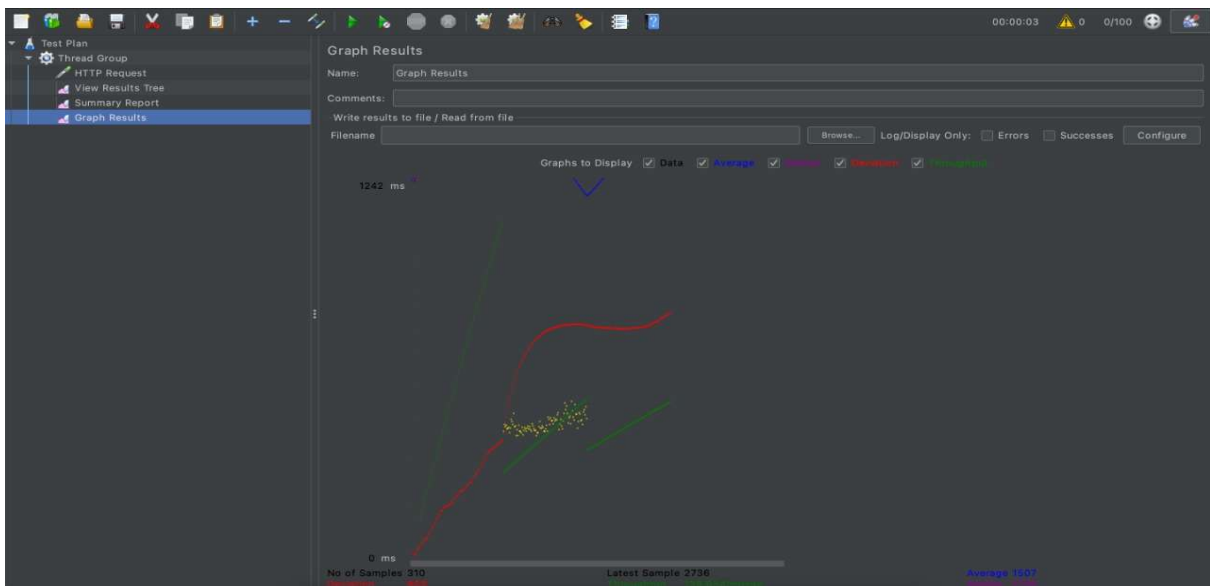


Figura 42. Imagen de línea de tiempo de la herramienta JMeter

### 3.5.4 Fase de Despliegue

Tal como indica la propuesta metodológica la fase de despliegue, es una fase importante dentro del proceso que tiene el desarrollo del software. En esta parte abordaremos un plan de despliegue, el mismo que consta de una serie de acciones relacionadas entre sí, estas actividades pueden estar de cualquiera de los dos lados como son de desarrollador, así como del lado del usuario, esto se debe a que cada proyecto de desarrollo es único, por ese motivo los procesos y actividades difícilmente pueden definirse. Las actividades que la propuesta metodológica sugiere se basan en el siguiente ciclo de vida.



Figura 43. Ciclo de vida propuesto para la etapa de despliegue, elaboración propia del autor

- a. Entorno Planificado; las actividades de seguridad implican muchas actividades, una planificación correcta puede definir el éxito y el fracaso en un proyecto, las actividades de preparación del entorno incluyen al sistema operativo, servidor web y el lenguaje de programación, desde la perspectiva de seguridad, se requiere saber la configurada del software, la disminución de los falsos positivos y garantizar que el nivel de detección vulnerabilidades sea adecuado.
- b. Configuraciones y pruebas ejecutadas; aquí todos los componentes están listos para ser ejecutados en un ambiente de pruebas, el cual tendrá las mismas configuraciones como las del producto final
- c. Correcciones; aquí participan todos los implicados en el desarrollo del proyecto, en esta parte de podrá detectar los errores y corregirlos antes de pasar a la etapa de producción.

- d. Producción; las fases descritas en líneas que anteceden a este punto se las puede ejecutar de forma constante, hasta conseguir el estado esperado de tener un producto seguro, en esta subsección se logra, activar o modificar los controles que no se habían hecho, en etapas anteriores como ajustes de firewalls, activaciones de copias de seguridad del software entre otros.

## CAPÍTULO 4 DISCUSIÓN DE LOS RESULTADOS OBTENIDOS EN EL ESTUDIO REALIZADO Y SU CORROBORACIÓN

En esta sección del proyecto se analizan los resultados que permiten aprobar la implementación de la Propuesta Metodológica de Desarrollo Ágil de Software con Énfasis en la Seguridad, en la empresa SOLNUS de la ciudad de Loja, desarrollando la iteración Migración De Tecnologías De La Plataforma. Los datos que se han recopilado se los obtuvo mediante la técnica de la entrevista, estos datos nos permiten a su vez corroborar que los objetivos planteados y la hipótesis de la investigación sean aprobados.

Las entrevistas fueron dirigidas a dos grupos de personas que se detallan a continuación:

- Departamento técnico de la empresa SOLNUS (Desarrolladores)
- Personal administrativo de la empresa SOLNUS (Testing)

### 4.1 Procesamiento de Datos y Corroboración de Resultados

La valoración está dada de la siguiente manera:

- Valor 1 corresponde a Difícil
- Valor 2 corresponde a Normal
- Valor 3 corresponde a Fácil / Optimo

La entrevista fue aplicada a los desarrolladores de le empresa SOLNUS, La información receptada será de carácter confidencial, la duración de esta entrevista será de 20 minutos aproximadamente. Cada pregunta tendrá un valor a fin de que el mismo nos permita interpretar los resultados en cuadros estadísticos. La Tabla 26, describe la valoración de cada pregunta, para mayor comprensión las preguntas se encuentran en los anexos.

PREGUNTA	VALOR		
	DIFICIL (1)	NORMAL (2)	FACIL/OPTIMO (3)
-----			

*Tabla 26. Valoración de preguntas, elaboración propia del autor*

- Para la interpretación de resultados se sumará todas las X de las respuestas de las preguntas realizadas
- El resultado total o suma de las X, se multiplicará por el valor que corresponda a cada columna, de esta forma se obtiene el resultado final
- Una vez obtenidos los resultados finales producto de la multiplicación se los divide por la sumatoria de las columnas
- La entrevista será estructurada teniendo una secuencia ordenada de preguntas, esta dividida en tres secciones: la primera sección, enfocada a evaluar la necesidad y aceptación de la propuesta metodológica, la segunda sección, orientada a la adaptación de los desarrolladores con el uso de la propuesta y la tercera parte, dirigida a dar opiniones para mejorar en un trabajo futuro la propuesta metodológica. Las primeras ocho preguntas están enfocadas al valor de la variable independiente y las cinco restantes para la variable dependiente.

Para la interpretación de los resultados, el tesista, así como su tutor, establecieron tres rangos de puntaje sobre 100%, quedando de la siguiente manera como se describe en la Tabla 27:

Rango	Descripción	Valor Mínimo	Valor Máximo	Cumple/No cumple
1	Si el valor comprende entre el mínimo y máximo de este rango la propuesta metodológica, no desempeña su objetivo, cumple en ciertos aspectos o su aceptación es muy baja por los entrevistados.	0 %	40%	No Cumple
2	Si el valor comprende entre el mínimo y máximo de este rango la propuesta metodológica, desempeña su objetivo, con algunas falencias y condiciones con relación al atributo de seguridad.	41%	60 %	Cumple con falencias
3	Si el valor comprende entre el mínimo y máximo de este rango la propuesta metodológica, desempeña su objetivo, como un aporte aplicativo en el desarrollo de software con énfasis en la seguridad	61 %	100 %	Cumple

Tabla 27. Rangos de valoración de Cumplimiento y no Cumplimiento, elaboración propia del autor



#### 4.2 Procesamiento Y Análisis para Comprobar la Necesidad y Aceptación de la Propuesta Metodológica con Énfasis en la Seguridad

En esta sección mostramos el análisis de los datos recopilados con el instrumento de recolección como es la entrevista, para de esta manera comprobar la necesidad y aceptación de la propuesta metodológica, para esta actividad se contó con el equipo de desarrollo durante la implementación de la propuesta.

La técnica de la entrevista se realizó en un grupo conformado por 4 personas, este procesamiento de datos se muestra a continuación en la Tabla 28:

Crterios	Difícil	Estándar	Fácil
Valores	2	3	4
Aciertos	3	20	29
Multiplicación	*2	*3	*4
Resultado parcial	6	60	116
Total	182		
FORMULA [(Total(#p52*3)) *100%]	87,50		

Tabla 28. Resultados previa a la aplicación de la Propuesta Metodológica (Personal Técnico empresa SONUS, Loja), elaboración propia del autor

Donde:

Total: suma total

#p: número total de preguntas

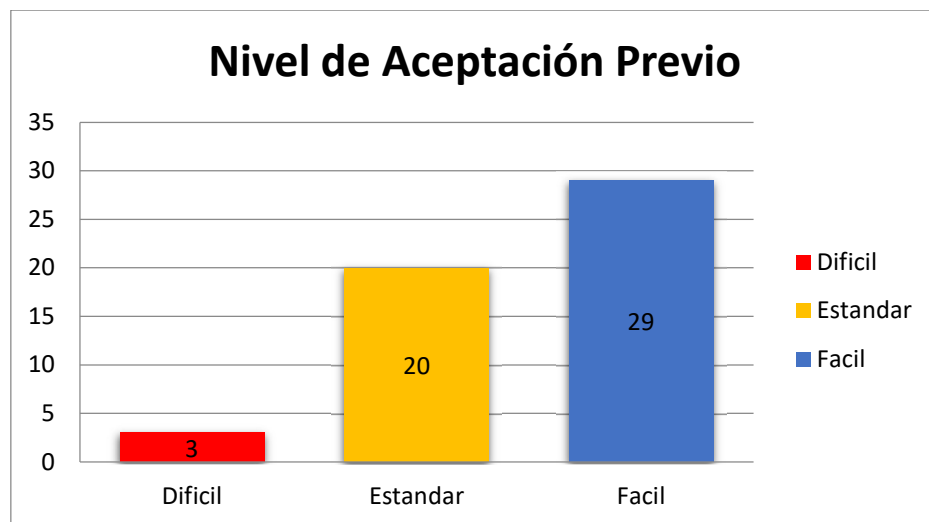


Figura 44. Representación Gráfica de los niveles de aceptación, previa a la aplicación de la Propuesta Metodológica (empresa SOLNUS, Loja), elaboración propia del autor

Para el cálculo de los valores se utilizó la siguiente fórmula: la sumatoria total de puntos alcanzados dividido para el producto de la multiplicación de el número total de preguntas por el valor más alto de acuerdo a los niveles, y todo esto se multiplicó por 100. Los resultados obtenidos luego de realizar la entrevista 1, previa a la implementación, se alcanzó una aceptación del 87,50%, de esta manera podemos definir que la aceptación y necesidad de usar una propuesta metodológica con énfasis en seguridad es necesaria, ya que permitirá subir los niveles de seguridad en los proyectos de desarrollados y a su vez sea adaptable a cualquier tipo de proyecto.

#### **4.3 Procesamiento y Análisis para Comprobar la Aceptación de las Técnicas, Métodos y Buenas Prácticas de Seguridad de la Propuesta Metodológica con Énfasis en la Seguridad**

En esta sección mostramos el análisis de los datos recopilados con el instrumento de recolección como es la entrevista, para de esta manera comprobar la aceptación de las técnicas, métodos y buenas prácticas de seguridad de la propuesta metodológica, para esta actividad se contó con el mismo equipo de desarrollo durante la implementación de la propuesta.

La técnica de la entrevista se realizó en el mismo grupo conformado por 4 personas, este procesamiento de datos se muestra a continuación en la Tabla 29:

Crterios	Difícil	Normal	Fácil/Optimo
Valores	2	3	4
Aciertos	0	19	41
Multiplicación	*2	*3	*4
Resultado parcial	0	57	164
Total	221		
FORMULA $[(Total/(\#p60*3)) *100\%]$	92,08		

Tabla 29. Resultados de la experiencia de la Propuesta Metodológica (Personal Técnico empresa SONUS, Loja), elaboración propia del autor

Donde:

ST: suma total

#p: número total de preguntas

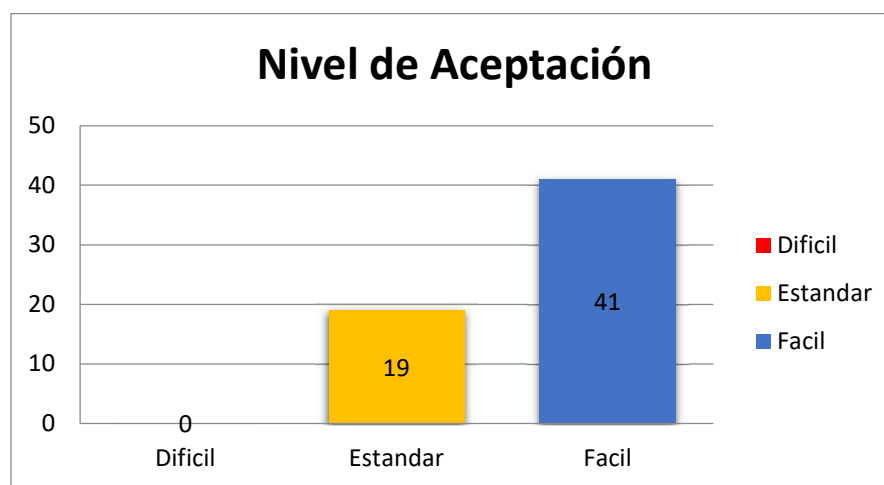


Figura 45. Representación Gráfica de los niveles de la experiencia de aceptación de las técnicas de la Metodología Propuesta, elaboración propia del autor

Para el cálculo de los valores se utilizó la siguiente formula: la sumatoria total de puntos alcanzados dividido para el producto de la multiplicación de el número total de preguntas

por el valor más alto de acuerdo a los niveles, y todo esto se multiplicó por 100. Los resultados obtenidos luego de realizar la entrevista, para medir el grado de aceptación de las técnicas, métodos y buenas prácticas de seguridad de la propuesta metodológica, se alcanzó una aceptación del 92,08%, de esta manera podemos definir que los desarrolladores les gustan la propuesta metodológica planteada para aumentar la seguridad en los proyectos que desarrollan, por tal motivo se puede decir que es adaptable a cualquier tipo de proyecto.

#### 4.4 Prueba de Hipótesis Chi Cuadrado

Para validar la hipótesis realizaremos la prueba de chi cuadrado, la misma que consiste en comparar los datos de la distribución observada con los datos de la distribución esperada. Esta comparación nos permite establecer una dependencia de una variable a la otra para determinar su cumplimiento. El detalle de las variables se muestra a continuación.

- a. **Hipótesis de la investigación:** Si se diseña la metodología de desarrollo ágil de software que agregue el atributo de seguridad, mediante la aplicación de técnicas de seguridades informáticas, entonces se minimizará las vulnerabilidades a los ataques informáticos
- b. **Variable Independiente:** Diseño de una propuesta metodológica de desarrollo de software ágil de software que agregue el atributo de seguridad, mediante la aplicación de técnicas de seguridades informáticas
- c. **Variable Dependiente:** Minimización de vulnerabilidades del software frente ataques informáticos
- d. **Hipótesis Nula (Ho):** El diseño de una propuesta metodológica de desarrollo ágil de software, con énfasis en la seguridad y las técnicas de seguridades informáticas son independientes.
- e. **Hipótesis alternativa (Ha):** El diseño de una propuesta metodológica de desarrollo ágil de software, con énfasis en la seguridad y las técnicas de seguridades informáticas son dependientes.

#### 4.4.1 Proceso de Obtención de las Frecuencias Esperadas y Observadas

En esta sección presentamos los datos que corresponden a las variables dependientes e independientes, estos datos fueron obtenidos de las entrevistas realizadas al equipo de desarrollo de la empresa SOLNUS de la ciudad de Loja y a los usuarios de los proyectos de dicha empresa, como se describe en las Tablas 30 y 31 respectivamente.

Valoración	Personal Desarrollo - Entrevista 1	Personal Desarrollo - Entrevista 2	Usuarios - Entrevista 3	Total
<b>Difícil</b>	1	0	0	<b>1</b>
<b>Estándar</b>	5	8	0	<b>13</b>
<b>Fácil</b>	10	16	0	<b>26</b>
<b>TOTAL</b>	<b>16</b>	<b>24</b>	<b>0</b>	<b>40</b>

Tabla 30. Propuesta Metodológica, Variable Independiente, elaboración propia del autor

Valoración	Personal Desarrollo - Entrevista 1	Personal Desarrollo - Entrevista 2	Usuarios - Entrevista 3	Total
<b>Difícil</b>	3	0	0	<b>3</b>
<b>Estándar</b>	16	11	20	<b>47</b>
<b>Fácil</b>	17	25	36	<b>78</b>
<b>TOTAL</b>	<b>36</b>	<b>36</b>	<b>56</b>	<b>128</b>

Tabla 31. Variable Dependiente: Minimización de vulnerabilidades del software frente ataques informáticos, elaboración propia del autor

##### 4.1.1.1 Frecuencia Esperada

La frecuencia esperada, es el conteo de observaciones que se espera de una celda. En la Tabla siguiente se indica los cruces de variables entre la Propuesta Metodológica y la Minimización de Vulnerabilidades del software frente ataques informáticos, la formula dada esta especificada por la teoría que nos indica como se da el resultado de esta frecuencia.

$$E_{ij} = \frac{\sum_{i=1}^m O_{i,j} * \sum_{j=1}^n O_{i,j}}{\sum_{i=1}^m \sum_{j=1}^n O_{i,j}}$$

Componentes:

m: total de columnas

n: total de filas

j: punto en la columna

i: punto en la fila

O: Observada

E: Esperada

Esta investigación es de fines exploratorios, se aprovechó las utilidades que nos brinda el programa IBM SPSS v25, realizándose el cruce de las variables y se obtuvo la siguiente Tabla 32, que nos describe este resultado:

			Variables de la Investigación		
			V.D. Minimización de vulnerabilidades del software frente ataques informáticos	V.I. Propuesta Metodológica	Total
<b>Valores</b>	Difícil	F. Observada	3	2	5
		F. Esperada	3,83	1,17	5,00
<b>Valores</b>	Fácil / Optimo	F. Observada	78	24	102
		F. Esperada	78,18	23,82	102,00
<b>Valores</b>	Normal	F. Observada	47	13	60
		F. Esperada	45,99	14,01	60,00
<b>TOTAL</b>		F. Observada	128	39,00	167
		F. Esperada	128,00	39,00	167,00

Tabla 32. Frecuencia Observada y Esperada según SPSS v25, elaboración propia del autor

#### 4.1.1.2 Cálculos Estadísticos del Chi Cuadrado Observado

Para este proceso, se utiliza el programa especializado en cálculos estadísticos como lo es SPSS v25, el cual al cruzar las variables Independiente y Dependiente nos muestra el siguiente resultado, descrito en la Tabla 33:

PRUEBA DE CHI CUADRADO	
Chi Cuadrado de Pearson	6,000 <sup>a</sup>

Tabla 33. Pruebas de Chi Cuadrado, elaboración propia del autor

El resultado que nos arroja el programa después de utilizar correctamente las variables en las tablas cruzadas es de 6,000<sup>a</sup>, expresado  $X^2$  Observado = 6,000<sup>a</sup>.

#### 4.1.1.3 Valor Calculado con la Técnica de chi cuadrado

Dentro de los pasos que se siguen para la corroboración de la hipótesis el siguiente calculo a seguir es la obtención del valor supuesto de significancia. Este valor se lo calcula utilizando la siguiente formula:

$$Vgl = (Cantidad\ de\ filas - 1) * (Cantidad\ de\ columnas - 1)$$

Donde: Vgl es el valor de grados de libertad.

La Tabla de valores observados tiene un total de filas y columnas de 3, expresado en la formula quedaría de la siguiente manera:

$$Vgl = (3-1) * (3-1)$$

$$Vgl = 2 * 2$$

$$Vgl = 4$$

El valor de grados de libertad será de 4, el nivel de significancia supuesto será de 0,05. En la siguiente Tabla 31, se describe la distribución de valores que esta técnica de chi cuadrado tiene, la intersección de los dos puntos nos determina el valor crítico, en este caso el valor corresponde a 9,4877.

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581
7	24,3213	22,0402	20,2777	18,4753	16,0128	14,0671	12,0170	10,7479	9,8032
8	26,1239	23,7742	21,9549	20,0902	17,5345	15,5073	13,3616	12,0271	11,0301
9	27,8767	25,4625	23,5893	21,6660	19,0228	16,9190	14,6837	13,2880	12,2421
10	29,5879	27,1119	25,1881	23,2093	20,4832	18,3070	15,9872	14,5339	13,4420

Tabla 31. Pruebas de Chi Cuadrado

#### 4.1.1.4 Comparativa entre los Valores Observados Y Críticos

Esta operación de comparar, nos ayuda a verificar la eficacia de la hipótesis nula, en la gráfica siguiente observamos que la observada tiene el valor de 9,4877, se encuentra en la zona de rechazo, en cambio la hipótesis nula tiene un valor de = 6,000, para mayor ejemplarización se muestra la siguiente grafica

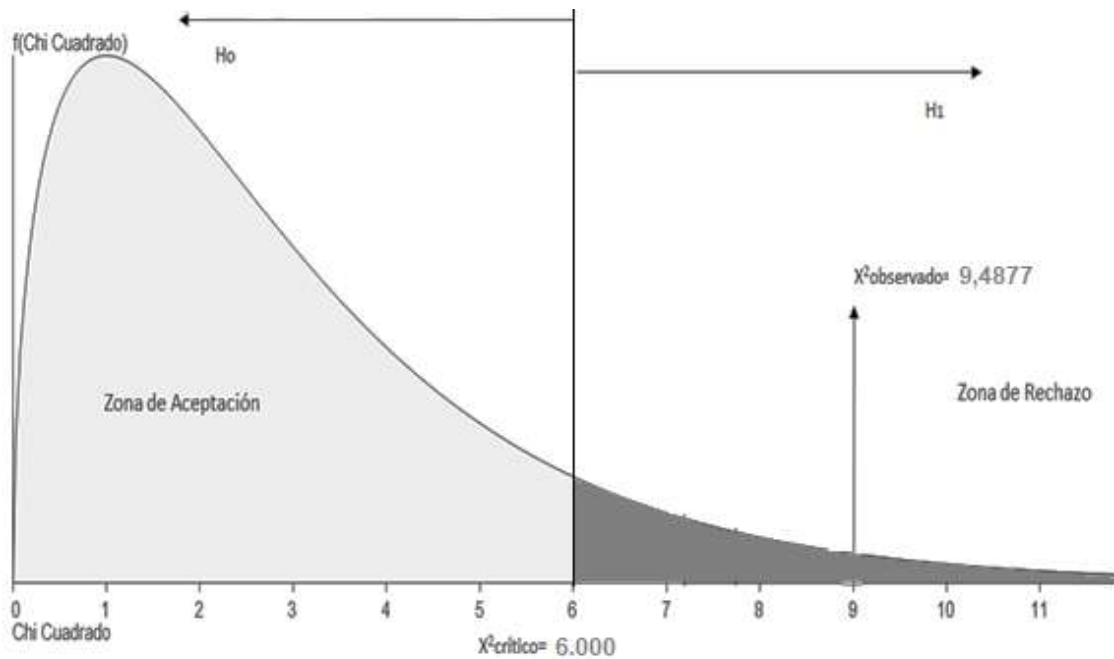


Figura 46. Grafico del Chi Cuadrado



Para el análisis del gráfico se tomaron en cuenta las reglas de decisión que inciden en la comparativa que se hace entre los valores observado y crítico, de esta manera las reglas aprueban el desempeño de la hipótesis.

- La  $H_0$  o Hipótesis Nula se acepta solo si:  $X^2 \text{ Observado} < X^2 \text{ Critico}$
- La Hipótesis de la Investigación se acepta solo si:  $X^2 \text{ Observado} > X^2 \text{ Critico}$

Para este caso de estudio se acepta la hipótesis de la investigación ya que cumple la regla de decisión estipulada, dado que el valor  $9,4877 > 6,000$

## CONCLUSIONES

- Este trabajo investigativo hace referencia a las técnicas, métodos y buenas prácticas de seguridad que han permitido durante largo tiempo, que las interfaces tengan mayor interactividad entre los usuarios y los sistemas en el desarrollo de software ágil.
- Las técnicas de seguridad aplicadas en esta investigación, mantienen semejanza con las propuestas de la ingeniería de software, esto permitió la integración de las mismas en cada fase del ciclo de vida propuesto, permitiendo la adaptación de la propuesta metodológica.
- El mayor problema que existió en toda la investigación, fue el de querer añadir las técnicas y sus entregables, dentro del ciclo de vida propuesto, esto ocasiono el retraso de tiempos previstos para los entregables en cada fase, pero gracias a la ayuda de las guías de buenas prácticas dadas por el proyecto OWASP y de la misma ingeniería de requisitos de seguridad, ayudo a que no se agreguen actividades ni entregables al mismo ciclo, sino más bien mejorar las existentes.
- Las pruebas estadísticas que se realizaron en la empresa SOLNUS de la ciudad de Loja, al aplicar la propuesta metodológica, permitieron concluir con una optimización de la seguridad, proporcionando de esta manera un producto confiable para su entrega final.
- Se concluye diciendo que los objetivos planteados se cumplieron en su totalidad ya que el producto de esta investigación fue validado con éxito en un caso específico dentro de una empresa de desarrollo de software, cumpliendo así con el objetivo principal el cual es el diseño de una metodología con énfasis en la seguridad.

## RECOMENDACIONES

- Para la realización de futuras aportes en base a esta investigación, se recomienda ahondar las técnicas, métodos y buenas prácticas de seguridad, además de las metodologías y los ciclos de vida que propone cada una de las mismas, ya que el uso de estas técnicas indica que sirven correctamente en el campo empírico.
- Utilizar la propuesta metodológica, ya que optimiza la percepción del desarrollador sobre la seguridad que debe tener en cuenta en cada fase del ciclo del software, que tiene una iteración ágil en el desarrollo del software.
- Para la implementación de la propuesta metodológica, se recomienda seguir detalladamente los pasos indicados, aplicar las técnicas, métodos y buenas prácticas de seguridad ya que han sido probadas y validadas en un entorno real, comprobando así su efectividad.
- Se recomienda utilizar una guía de trazabilidad a fin de que esta nos sirva de guía completa para saber qué información debemos ingresar y que información vamos a desarrollar.

## BIBLIOGRAFIA

- [1] Laskowski, «Agile IT Security Implementation Methodology - Google Libros», 2011. <https://books.google.es/books?hl=es&lr=&id=sMtZ3lBfmmMC&oi=fnd&pg=PT9&dq=Agile+IT+Security+Implementation+Methodology&ots=hx9fT8NXdo&sig=iYjngxtWwNcxXOwfuCjqJRtGg#g#v=onepage&q=Agile%20IT%20Security%20Implementation%20Methodology&f=false> (accedido jul. 05, 2020).
- [2] «CiberSeguridad». <https://observatoriociberseguridad.org/#/home> (accedido feb. 19, 2021).
- [3] C. A. del P. Anchundia, «Evaluación del plan de seguridad informática diseñado para el Tecnológico de la Universidad Laica Eloy Alfaro de Manabí», *Rev. UNIANDÉS Episteme*, vol. 4, n.º 2 (abril-junio), pp. 263-275, 2017.
- [4] Senplades, «S. Plan Nacional de Desarrollo 2017–2021 Toda una Vida.», 2017. <https://www.planificacion.gob.ec/> (accedido jul. 05, 2020).
- [5] MINTEL, «Más de 40 millones de ataques al Ecuador neutralizados desde el retiro del asilo a Julian Assange – Ministerio de Telecomunicaciones y de la Sociedad de la Información», 2019. <https://www.telecomunicaciones.gob.ec/mas-de-40-millones-de-ataques-al-ecuador-neutralizados-desde-el-retiro-del-asilo-a-julian-assange/> (accedido jun. 27, 2020).
- [6] «Un informe de INTERPOL muestra un aumento alarmante de los ciberataques durante la epidemia de COVID-19». <https://www.interpol.int/es/Noticias-y-acontecimientos/Noticias/2020/Un-informe-de-INTERPOL-muestra-un-aumento-alarmante-de-los-ciberataques-durante-la-epidemia-de-COVID-19> (accedido feb. 15, 2021).
- [7] Marulanda, C. y Ceballos, J, «Una revisión de metodologías seguras en cada fase del ciclo de vida del desarrollo de software | Ingenierías USBMed», 2012. <http://190.131.242.67/index.php/IngUSBmed/article/view/260> (accedido jul. 05, 2020).
- [8] Awad, M. A., «A comparison between agile and traditional software development methodologies. », *Univ. West. Aust.*, vol. 30, p. 84, 2005.
- [9] P. Hohl et al., «Back to the future: origins and directions of the “Agile Manifesto” – views of the originators», *J. Softw. Eng. Res. Dev.*, vol. 6, n.º 1, p. 15, nov. 2018, doi: 10.1186/s40411-018-0059-z.
- [10] J. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Addison-Wesley, 2013.
- [11] A. Cockburn, *Agile Software Development: The Cooperative Game*. Pearson Education, 2006.
- [12] J. Stapleton, *Dsdm: The Method in Practice*. USA: Addison-Wesley Longman Publishing Co., Inc., 1997.
- [13] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2000.
- [14] P. Coad, J. de Luca, y E. Lefebvre, *Java Modeling Color with Uml: Enterprise Components and Process with Cdrom*, 1st ed. USA: Prentice Hall PTR, 1999.
- [15] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004.
- [16] Z. Azham, Ghani & Ithnin, «Security backlog in Scrum security practices», en *2011 Malaysian Conference in Software Engineering*, 2011, pp. 414-417, doi: 10.1109/MySEC.2011.6140708.
- [17] Graham Cheetham y G. E. Chivers, «Professions, Competence and Informal Learning - Google Libros», 2005. <https://books.google.es/books?hl=es&lr=&id=xwyyqLR-mG4cC&oi=fnd&pg=PR7&dq=Chivers,+2005&ots=XFIRfFGZag&sig=OgJVvcnJ30b>

- WO3zIn\_Q2dIY6R2I#v=onpage&q=Chivers%2C%202005&f=false (accedido jul. 05, 2020).
- [18] V. Casola, A. D. Benedictis, M. Rak, y U. Villano, «A methodology for automated penetration testing of cloud applications», *Int. J. Grid Util. Comput.*, vol. 11, n.º 2, p. 267, 2020, doi: 10.1504/IJGUC.2020.105541.
- [19] A. L. Mesquida y A. Mas, «Implementing information security best practices on software lifecycle processes: The ISO/IEC 15504 Security Extension», *Comput. Secur.*, vol. 48, pp. 19-34, feb. 2015, doi: 10.1016/j.cose.2014.09.003.
- [20] R. S. Pressman y B. R. Maxim, *Software Engineering: A practitioner's approach*, vol. 8th ed. New York, NY, USA: McGraw-Hill Education, 2015.
- [21] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, y S. Linkman, «Systematic literature reviews in software engineering – A systematic literature review», *Inf. Softw. Technol.*, vol. 51, n.º 1, pp. 7-15, ene. 2009, doi: 10.1016/j.infsof.2008.09.009.
- [22] H. Altunel, «Agile Project Management in Product Life Cycle», *Int. J. Inf. Technol. Proj. Manag. IJITPM*, vol. 8, n.º 2, pp. 50-63, 2017, doi: 10.4018/IJITPM.2017040104.
- [23] A. Buchalcevova, «Application of Methodology Evaluation System on Current IS Development Methodologies», *Int. J. Inf. Technol. Syst. Approach IJITSA*, vol. 11, n.º 2, pp. 71-87, 2018, doi: 10.4018/IJITSA.2018070105.
- [24] T. Gu, M. Lu, L. Li, y Q. Li, «An Approach to Analyze Vulnerability of Information Flow in Software Architecture», *Appl. Sci.*, vol. 10, n.º 1, p. 393, ene. 2020, doi: 10.3390/app10010393.
- [25] A. Shah, K. A. Farris, R. Ganesan, y S. Jajodia, «Vulnerability Selection for Remediation: An Empirical Analysis»: *J. Def. Model. Simul.*, sep. 2019, doi: 10.1177/1548512919874129.
- [26] A. Javan Jafari y A. Rasoolzadegan, «Quality-centric security pattern mutations», *Softw. Qual. J.*, vol. 27, n.º 4, pp. 1531-1561, dic. 2019, doi: 10.1007/s11219-019-09454-5.
- [27] Y. Pan et al., «Detecting web attacks with end-to-end deep learning», *J. Internet Serv. Appl.*, vol. 10, n.º 1, p. 16, ago. 2019, doi: 10.1186/s13174-019-0115-x.
- [28] H. García, «Metodología de la Investigación 5a Edición - Sampieri, Fernandez copia», Accedido: nov. 07, 2020. [En línea]. Disponible en: [https://www.academia.edu/36064159/Metodologia\\_de\\_la\\_Investigacion\\_5a\\_Edici%C3%B3n\\_Sampieri\\_Fernandez\\_copia](https://www.academia.edu/36064159/Metodologia_de_la_Investigacion_5a_Edici%C3%B3n_Sampieri_Fernandez_copia).
- [29] M. E. de la Mora, *Metodología de la investigación: desarrollo de la inteligencia*. International Thomson Editores, S.A. de C.V., 2006.
- [30] J. R. M. Ríos, M. P. Z. Ordóñez, M. J. C. Segarra, y F. G. G. Zerda, «Estado del arte: Metodologías de desarrollo en aplicaciones web», *3c Technol. Glosas Innov. Apl. Pyme*, vol. 6, n.º 3, pp. 54-71, 2017.
- [31] V. Balijepally, G. DeHondt, V. Sugumaran, y S. Nerur, «Agility in Software Development and Project Value: An Empirical Investigation», *J. Database Manag. JDM*, vol. 28, n.º 4, pp. 40-59, 2017, doi: 10.4018/JDM.2017100103.
- [32] J. R. M. Ríos, M. P. Z. Ordóñez, M. J. C. Segarra, y F. G. G. Zerda, «Comparación de metodologías en aplicaciones web», *3c Technol. Glosas Innov. Apl. Pyme*, vol. 7, n.º 1, pp. 1-19, 2018.
- [33] M. D. Freitas, F. C. C. Araujo, y S. L. B. França, «COMPARATIVE ANALYSIS OF PROJECT MANAGEMENT METHODOLOGIES PMBOK AND AGILE – A CASE STUDY WITH COMPANIES OF THE BRAZILIAN ENERGETIC SECTOR», *Rev. Gest. Inov. E Technol.*, vol. 9, n.º 3, jul. 2019, doi: 10.7198/geintec. v9i3.1340.
- [34] R. Matamoros y G. Lilibet, «Auditoria de seguridad informática en los laboratorios de la Unidad Académica de Ciencias Empresariales de la UTMACH.», 2018, Accedido:

- nov. 07, 2020. [En línea]. Disponible en: <http://repositorio.utmachala.edu.ec/handle/48000/12985>.
- [35] M. Castellaro, S. C. Romaniz, y P. A. Pessolani, «Hacia la Ingeniería de Software Seguro», presentado en XV Congreso Argentino de Ciencias de la Computación, 2009, Accedido: nov. 07, 2020. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/21332>.
- [36] G. McGraw, «Software security», *IEEE Secur. Priv.*, vol. 2, n.º 2, pp. 80-83, mar. 2004, doi: 10.1109/MSECP.2004.1281254.
- [37] L. Ardila y K. Yojana, «Metodologías ágiles como herramientas fundamentales para el desarrollo de emprendimientos.», abr. 2020, Accedido: nov. 17, 2020. [En línea]. Disponible en: <http://repository.unad.edu.co/handle/10596/33613>.
- [38] D. Mellado, E. Fernández-Medina, y M. Piattini, «A common criteria-based security requirements engineering process for the development of secure information systems», *Comput. Stand. Interfaces*, vol. 29, n.º 2, pp. 244-253, feb. 2007, doi: 10.1016/j.csi.2006.04.002.
- [39] N. R. Mead y T. Stehney, «Security quality requirements engineering (SQUARE) methodology», *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, n.º 4, pp. 1-7, may 2005, doi: 10.1145/1082983.1083214.
- [40] J. Jürjens, «Foundations for Designing Secure Architectures», *Electron. Notes Theor. Comput. Sci. ENTCS*, vol. 142, pp. 31-46, ene. 2006, doi: 10.1016/j.entcs.2005.07.012.
- [41] M. Nicho, «A process model for implementing information systems security governance», *Inf. Comput. Secur.*, vol. 26, n.º 1, pp. 10-38, mar. 2018, doi: 10.1108/ICS-07-2016-0061.
- [42] N. R. Mead, J. H. Allen, S. Barnum, R. J. Ellison, y G. R. McGraw, *Software Security Engineering: A Guide for Project Managers*. Addison-Wesley Professional, 2004.
- [43] M. Alenezi, A. Agrawal, R. Kumar, y R. A. Khan, «Evaluating Performance of Web Application Security Through a Fuzzy Based Hybrid Multi-Criteria Decision-Making Approach: Design Tactics Perspective», *IEEE Access*, vol. 8, pp. 25543-25556, 2020, doi: 10.1109/ACCESS.2020.2970784.
- [44] S. Ali, Y. Hafeez, S. Hussain, y S. Yang, «Enhanced regression testing technique for agile software development and continuous integration strategies», *Softw. Qual. J.*, vol. 28, n.º 2, pp. 397-423, jun. 2020, doi: 10.1007/s11219-019-09463-4.
- [45] «OWASP Top 10 Security Vulnerabilities 2020», *Sucuri*. <https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/> (accedido nov. 07, 2020).
- [46] M. C. Sanchez, J. M. C. de Gea, J. L. Fernandez-Aleman, J. Garceran, y A. Toval, «Software vulnerabilities overview: A descriptive study», *Tsinghua Sci. Technol.*, vol. 25, n.º 2, pp. 270-280, abr. 2020, doi: 10.26599/TST.2019.9010003.
- [47] J. Akram y L. Ping, «How to build a vulnerability benchmark to overcome cyber security attacks», *IET Inf. Secur.*, vol. 14, n.º 1, pp. 60-71, ene. 2020, doi: 10.1049/iet-ifs.2018.5647.
- [48] G.-Y. Chan, F.-F. Chua, y C.-S. Lee, «Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns», *J. Intell. Fuzzy Syst.*, vol. 31, n.º 2, pp. 749-764, ene. 2016, doi: 10.3233/JIFS-169007.
- [49] de Vicente Mohino, Bermejo Higuera, Bermejo Higuera, y Sicilia Montalvo, «The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies», *Electronics*, vol. 8, n.º 11, p. 1218, oct. 2019, doi: 10.3390/electronics8111218.

- [50] Fahad, Muhammad; Qadri, Salman; Ullah, Saleem; Husnain, Mujtaba; Qaiser, Rizwan; Qureshi, Shehzad Ahmed; Ahmed, Waqas; Muhammad, Syed Shah, «A Comparative Analysis of DXPRUM and DSDM», *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, n.º 5, pp. 259-264, may 2017.
- [51] S. M. Ghaffarian y H. R. Shahriari, «Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques: A Survey», *ACM Comput. Surv.*, vol. 50, n.º 4, pp. 1-36, nov. 2017, doi: 10.1145/3092566.
- [52] S. N. G. Gourisetti, M. Mylrea, y H. Patangia, «Cybersecurity vulnerability mitigation framework through empirical paradigm: Enhanced prioritized gap analysis», *Future Gener. Comput. Syst.*, vol. 105, pp. 410-431, abr. 2020, doi: 10.1016/j.future.2019.12.018.
- [53] C. Chipantiza y V. Lewis, «Desarrollo de un marco de adaptación de la ingeniería de la usabilidad al proceso de desarrollo Ágil SCRUM, aplicado en el Departamento de Planificación del ECU911 de la Ciudad de Machala.», jul. 2015, Accedido: nov. 18, 2020. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/jspui/handle/21000/10119>.
- [54] «OWASP Foundation | Open-Source Foundation for Application Security». <https://owasp.org/> (accedido nov. 07, 2020).
- [55] *Guardsquare/proguard*. Guardsquare, 2020.

## ANEXOS

### Anexo 1

# <Nombre Proyecto>

## Plan de Pruebas

Versión: 0100

Fecha: DD/MM/AAAA

### INTRODUCCIÓN

#### Objeto

El objetivo de este documento es recoger los casos de pruebas que verifican que el sistema satisface los requisitos especificados. Deberá contener la definición de los casos de prueba, la matriz de trazabilidad entre casos de pruebas y requisitos, y la estrategia a seguir en la ejecución de las pruebas.

#### Alcance

Unidades organizativas y responsabilidades a las que va dirigida el documento

### TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS

En este apartado se deberá completar una matriz como la que se indica a continuación, en la cual se indicará la correspondencia entre los casos de pruebas definidos y los requisitos de seguridad de la especificación de requisitos. Las filas representan cada uno de los casos de pruebas definidos, y las columnas los requisitos funcionales. Si un caso de prueba se encarga de verificar un requisito, se tendrá que señalar con una X la casilla correspondiente.

	Requisitos funcionales de seguridad 01	Requisitos funcionales de seguridad 02	Requisitos funcionales de seguridad 03	Requisitos funcionales de seguridad 04	Requisitos funcionales de seguridad...n
Control de pruebas 01	X				X
Control de pruebas 02		X	X	X	
Control de pruebas 03	X	X			X



Control de pruebas...n			x		x
Control de pruebas...n	x			x	

### DEFINICIÓN DE LOS CASOS DE PRUEBAS

En este apartado se describirán en detalle cada uno de los casos de pruebas que se hayan identificado como necesarios para verificar la funcionalidad completa del sistema. Se deberá repetir una tabla por cada caso de prueba que se defina. Del conjunto de casos de pruebas definidos, deberán identificarse aquellos que formarán parte del conjunto de pruebas que deberán realizarse para asegurar el correcto despliegue de la aplicación.

<Nombre caso prueba>	<Código del CP>	
	¿Prueba de despliegue?	Si/No
Descripción: <Descripción del caso de prueba>		
Prerrequisitos <Enumerar los prerrequisitos para la prueba>		
Pasos: <Pasos generales para la prueba, basados en los escenarios de los casos de uso, si existen.>		
Resultado esperado: <Resultado esperado de la prueba>		
Resultado obtenido: <Resultado obtenido de la ejecución del caso de prueba>		

## Anexo 2

# <Nombre Proyecto> Plan de Respuesta a Incidentes

Versión: 0100

Fecha: DD/MM/AAAA

### Objetivo

El objetivo de este documento es recoger los incidentes que se dieron durante la ejecución del proyecto, describir el incidente, el impacto generado entre otros puntos importantes.

### Alcance

Unidades organizativas y responsabilidades a las que va dirigida el documento

Para este plan debemos llenar la siguiente matriz con la información referente como se indica en cada sección de la misma, este formulario nos ayudara a llevar un control de todos los incidentes que se generaron en el desarrollo del ciclo de vida del software, se deberá generar una matriz por cada incidente encontrado.

PLAN DE RESPUESTA A INCIDENTES CODIGO: PRI01...			
PROYECTO			
GERENTE			
PREPARADO POR:		FECHA	
REVISADO POR		FECHA	
NUMERO DEL INCIDENTE			
DENOMINACION DEL INCIDENTE			
<b>1. DESCRIPCION DE INCIDENTE</b> (Aquí se describe lo ocurrido, cuáles fueron las causas, quienes participaron, reacciones y efectos inmediatos)			
<b>2. IMPACTO QUE PODRIA GENERAR EL INCIDENTE</b> (Como afecta el incidente ocurrido a los objetivos del proyecto)			

<b>3. ROLES INVOLUCRADOS EN EL INCIDENTE</b> (Interesados en el proyecto que fueron partícipes indirectos del incidente)			
NOMBRES APELLIDOS	Y	ROL	ORGANIZACIÓN
<b>4. ACCIONES TOMADAS PARA RESOLVER EL INCIDENTE</b> (Estrategias, actividades o coordinaciones entre otras realizadas para resolver el incidente)			
<b>5. ACUERDOS TOMADOS PARA RESOLVER EL INCIDENTE</b> (Acuerdos, compromisos, etc. Tomados entre las partes formales y oficiales para resolver y superar el incidente)			
<b>6. FACILITADOR DEL INCIDENTE</b> (Nombres, apellidos, rol e información del contacto de la persona que actuó para remediar en primera instancia el incidente)			
<b>7. RECOMENDACIONES PARA FUTUROS PROYECTOS</b> (Pautas que deberían considerarse a futuro para evitar o minimizar que ocurran incidentes similares)			
<b>8. RELACION DE ANEXOS</b> (si aplica)			
ANEXO 1			
ANEXO 2			
ANEXO 3			
ANEXO N			

### Anexo 3

## <Nombre Proyecto>

### Plan de Riesgos

Versión: 0100

Fecha: DD/MM/AAAA

#### 1. Estrategia Global

Descripción de una aproximación en alto nivel sobre la gestión de riesgos del proyecto. Resumir cómo tienen que llevarse a cabo de forma colectiva las siguientes actividades de gestión de riesgos especificadas en el Plan de Gestión de Riesgos: identificación de los riesgos, análisis, priorización, respuesta, monitoreo, y control.

#### 2. Definición de Roles

Para cada uno de los roles y/o áreas funcionales del proyecto se deberá completar la matriz de actividades de gestión de riesgos. Una vez completa dicha matriz reflejará, para cada una de las actividades clave de la gestión de los riesgos, las responsabilidades asignadas a cada uno de los roles funcionales.

Actividad de Gestión del Riesgo	Rol 1	Rol 2						Rol N
Plan de Gestión del Riesgo para el desarrollo y administración								
Determinar si el Plan de Gestión de Riesgos está listo para ser aprobado								
Identificar los riesgos del proyecto								
Aprobar y autorizar el uso de Planes de Contingencia								
Leyenda C= responsabilidad compartida P= responsabilidad primaria S= responsabilidad de soporte								

#### 3. Evaluación del Riesgo

### **3.1 Identificación de los riesgos**

#### **3.1.1 Métodos y Técnicas**

Describir cómo se deben identificar y organizar los riesgos como preparación para el posterior análisis de los riesgos.

#### **3.1.2 Riesgos del Proyecto**

Identificar y describir los riesgos del proyecto que se utilizan como base para el análisis de riesgos. Identificar los riesgos específicos en función de los métodos y técnicas definidos.

### **4. Monitoreo y Control de los Riesgos**

#### **4.1 Seguimiento de los Riesgos**

Describir cómo el equipo del proyecto tiene que determinar si se está realizando de forma efectiva la gestión de los riesgos a lo largo del ciclo de vida completo del proyecto. La evaluación de la gestión de riesgos incluye de forma específica cómo se asegurará, para un determinado proyecto, que las acciones de respuesta al riesgo lo están manteniendo bajo control, incluyendo indicadores de monitoreo para conocer cuándo será necesario invocar a los planes de contingencia.

### **5. Reporte de los Riesgos**

#### **5.1 Ítems de Riesgo**

Describir métodos para reportar las actividades de mitigación de los riesgos. Describir el uso de informes para realizar de forma detallada la revisión y el seguimiento de las acciones de mitigación.

#### **5.2 Estado del Riesgo**

Describir métodos para reportar las actividades de mitigación de los riesgos. Describir el uso de informes para revisar y seguir las acciones de mitigación de los riesgos de forma detallada (como un único ítem del riesgo).

## Anexo 4

### Entrevista 1

Variables	Tipo entrevista	Entrevistados	Preguntas	Descripción	Respuestas
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	1	¿Cree usted necesario desarrollar una propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad para el Departamento de desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	2	¿Considera usted que, con la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, se optimice la seguridad en el proceso de desarrollo de software direccionado por el Departamento de desarrollo?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	3	¿Cree usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, abarca con todas las actividades de desarrollo de software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	4	¿Cree usted con la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, se optimice la calidad de la seguridad del software desarrollado?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	5	¿Cree usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad permitiría definir el alcance del proyecto a desarrollar?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	6	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad se entregaría al cliente un software completamente seguro?	Difícil

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	7	¿Considera usted que la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, cumple con los objetivos definidos por el departamento de desarrollo con respecto a la seguridad en el software?	Difícil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	8	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, considera usted que se puedan solucionar las falencias de seguridad en el desarrollo de software presentados al momento?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. José Bustillos	9	¿Considera usted que, con la Propuesta metodológica, es necesario aplicarlo como una metodología y estándar a seguir?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. José Bustillos	10	¿Considera usted que, con la Propuesta metodológica, sería fácil de ser aplicado por el personal técnico de la empresa SOLNUS de la ciudad de Loja?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. José Bustillos	11	¿Considera usted que con la Propuesta metodológica guarda los principios para el desarrollo ágil?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. José Bustillos	12	¿Considera usted que con la Propuesta metodológica promueve el trabajo en equipo entre el personal técnico y los usuarios?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. José Bustillos	13	¿Considera usted que las técnicas de seguridad de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad son de aprendizaje y aplicación fácil?	Difícil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	1	¿Cree usted necesario desarrollar un Marco de Adaptación entre Scrum y las técnicas de seguridad para el Departamento de desarrollo?	Estándar

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	2	¿Considera usted que, con seguridad para la Propuesta metodológica, se optimice la seguridad en el proceso de desarrollo de software direccionado por el Departamento de desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	3	¿Cree usted que con la Propuesta metodológica abarca con todas las actividades de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	4	¿Cree usted con que, la Propuesta metodológica se optimice la calidad de la seguridad del software desarrollado?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	5	¿Cree usted que con la Propuesta metodológica permitiría definir el alcance del proyecto a desarrollar?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	6	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad se entregaría al cliente un software completamente usable?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	7	¿Considera usted que con la Propuesta metodológica cumple con los objetivos definidos por el departamento de desarrollo con respecto a la seguridad en desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	8	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, considera usted que se puedan solucionar las falencias de seguridad en el desarrollo de software presentados al momento?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Nelson Agurto	9	¿Considera usted que, con la Propuesta metodológica, es necesario aplicarlo como una metodología y estándar a seguir?	Estándar
V.I. Propuesta Metodológica	Entrevista 1	Ing. Nelson Agurto	10	¿Considera usted que, con la Propuesta metodológica, sería fácil de ser aplicado por el personal técnico de la empresa SOLNUS de la ciudad de Loja?	Fácil



V.I. Propuesta Metodológica	Entrevista 1	Ing. Nelson Agurto	11	¿Considera usted que con la Propuesta metodológica guarda los principios para el desarrollo ágil?	Estándar
V.I. Propuesta Metodológica	Entrevista 1	Ing. Nelson Agurto	12	¿Considera usted que con la Propuesta metodológica promueve el trabajo en equipo entre el personal técnico y los usuarios?	Estándar
V.I. Propuesta Metodológica	Entrevista 1	Ing. Nelson Agurto	13	¿Considera usted que las técnicas de seguridad de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad son de aprendizaje y aplicación fácil?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	1	¿Cree usted necesario desarrollar un Marco de Adaptación entre Scrum y las técnicas de seguridad para el Departamento de desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	2	¿Considera usted que con que, la Propuesta metodológica, ¿se optimice la seguridad en el proceso de desarrollo de software direccionado por el Departamento de desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	3	¿Cree usted que con la Propuesta metodológica abarca con todas las actividades de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	4	¿Cree usted con seguridad para la Propuesta metodológica se optimice la calidad de la seguridad del software desarrollado?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	5	¿Cree usted que con la Propuesta metodológica permitiría definir el alcance del proyecto a desarrollar?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	6	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad se entregaría al cliente un software completamente seguro?	Estándar

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	7	¿Considera usted que con la Propuesta metodológica cumple con los objetivos definidos por el departamento de desarrollo con respecto a la seguridad en desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	8	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, considera usted que se puedan solucionar las falencias de seguridad en el desarrollo de software presentados al momento?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Michael Jiménez	9	¿Considera usted que, con la Propuesta metodológica, es necesario aplicarlo como una metodología y estándar a seguir?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. Michael Jiménez	10	¿Considera usted que, con la Propuesta metodológica, sería fácil de ser aplicado por el personal técnico de la empresa SOLNUS de la ciudad de Loja?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. Michael Jiménez	11	¿Considera usted que con la Propuesta metodológica guarda los principios para el desarrollo ágil?	Estándar
V.I. Propuesta Metodológica	Entrevista 1	Ing. Michael Jiménez	12	¿Considera usted que con la Propuesta metodológica promueve el trabajo en equipo entre el personal técnico y los usuarios?	Estándar
V.I. Propuesta Metodológica	Entrevista 1	Ing. Michael Jiménez	13	¿Considera usted que las técnicas de seguridad de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad son de aprendizaje y aplicación fácil?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	1	¿Cree usted necesario desarrollar un Marco de Adaptación entre Scrum y las técnicas de seguridad para el Departamento de desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	2	¿Considera usted que, con seguridad para la Propuesta metodológica, se optimice la seguridad en el proceso de desarrollo de software direccionado por el Departamento de desarrollo?	Estándar

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	3	¿Cree usted que con la Propuesta metodológica abarca con todas las actividades de desarrollo de software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	4	¿Cree usted con que, la Propuesta metodológica se optimice la calidad de la seguridad del software desarrollado?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	5	¿Cree usted que con la Propuesta metodológica permitiría definir el alcance del proyecto a desarrollar?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	6	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad se entregaría al cliente un software completamente usable?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	7	¿Considera usted que con la Propuesta metodológica cumple con los objetivos definidos por el departamento de desarrollo con respecto a la seguridad en desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	8	¿Con la aplicación de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad, considera usted que se puedan solucionar las falencias de seguridad en el desarrollo de software presentados al momento?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 1	Ing. Com. María Paz Ludeña	9	¿Considera usted que, con la Propuesta metodológica, es necesario aplicarlo como una metodología y estándar a seguir?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. Com. María Paz Ludeña	10	¿Considera usted que, con la Propuesta metodológica, sería fácil de ser aplicado por el personal técnico de la empresa SOLNUS de la ciudad de Loja?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. Com. María Paz Ludeña	11	¿Considera usted que con la Propuesta metodológica guarda los principios para el desarrollo ágil?	Fácil

V.I. Propuesta Metodológica	Entrevista 1	Ing. Com. María Paz Ludeña	12	¿Considera usted que con la Propuesta metodológica promueve el trabajo en equipo entre el personal técnico y los usuarios?	Fácil
V.I. Propuesta Metodológica	Entrevista 1	Ing. Com. María Paz Ludeña	13	¿Considera usted que las técnicas de seguridad de la propuesta metodológica de desarrollo de software ágil, con énfasis en la seguridad son de aprendizaje y aplicación fácil?	Fácil

## Anexo 5

### Entrevista 2

Variables	Tipo Entrevista	Entrevistados	Preguntas	Descripción	Respuestas
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	1	¿Cree usted que la Propuesta Metodológica cubre la necesidad de agregar el atributo de seguridad en las etapas del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	2	¿Considera usted que la Propuesta Metodológica cubre todos los aspectos necesarios del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	3	¿Cree usted que aplicando la Propuesta Metodológica se optimiza la seguridad en el proceso de desarrollo de software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	4	¿Cree usted que al aplicar la Propuesta Metodológica se lograría cumplir con los tiempos establecidos en la Desarrollo de un proyecto sin sacrificar la seguridad del software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	5	¿Cree usted que se podría entregar un software que satisfaga las aspiraciones del cliente y los usuarios, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	6	¿Cree usted que la Propuesta Metodológica cumple con los objetivos requeridos por el Departamento de Desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	7	¿Cree usted que la Propuesta Metodológica contribuye con controles para mejorar la Desarrollo establecida por el equipo de técnico?	Estándar

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	8	¿Cree usted que es posible solucionar las falencias de seguridad de software presentadas hasta el momento, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. José Bustillos	9	¿Considera usted que la Propuesta Metodológica mantiene una estructura lógica, organizada y estructurada para satisfacer el ciclo de vida del software?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	10	¿Considera usted que la Propuesta Metodológica se acopla ágilmente a cualquier proyecto de software que se desarrolla en el Departamento Desarrollo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	11	¿Cree usted que el esfuerzo utilizado por el personal técnico para aplicar la Propuesta Metodológica está en los parámetros normales?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	12	¿Considera usted que la Propuesta Metodológica se adapta a los requisitos cambiantes o si llegaran tarde al desarrollo de software?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	13	¿Considera usted que la Propuesta Metodológica fomenta el trabajo en equipo?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	14	¿Cree usted que se logrará la entrega a tiempo de los productos desarrollados bajo los parámetros de la seguridad con la aplicación de la Propuesta Metodológica?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. José Bustillos	15	¿Encuentra usted correcto el empleo de la Propuesta Metodológica?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	1	¿Cree usted que la Propuesta Metodológica cubre la necesidad de agregar el atributo de seguridad en las etapas del proceso de desarrollo de software?	Fácil

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	2	¿Considera usted que la Propuesta Metodológica cubre todos los aspectos necesarios del proceso de desarrollo de software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	3	¿Cree usted que aplicando la Propuesta Metodológica se optimiza la seguridad en el proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	4	¿Cree usted que al aplicar la Propuesta Metodológica se lograría cumplir con los tiempos establecidos en la Desarrollo de un proyecto sin sacrificar la seguridad del software?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	5	¿Cree usted que se podría entregar un software que satisfaga las aspiraciones del cliente y los usuarios, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	6	¿Cree usted que la Propuesta Metodológica cumple con los objetivos requeridos por el Departamento de Desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	7	¿Cree usted que la Propuesta Metodológica contribuye con controles para mejorar la Desarrollo establecida por el equipo de técnico?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	8	¿Cree usted que es posible solucionar las falencias de seguridad de software presentadas hasta el momento, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Nelson Agurto	9	¿Considera usted que la Propuesta Metodológica mantiene una estructura lógica, organizada y estructurada para satisfacer el ciclo de vida del software?	Estándar

V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	10	¿Considera usted que la Propuesta Metodológica se acopla ágilmente a cualquier proyecto de software que se desarrolla en el Departamento Desarrollo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	11	¿Cree usted que el esfuerzo utilizado por el personal técnico para aplicar la Propuesta Metodológica está en los parámetros normales?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	12	¿Considera usted que la Propuesta Metodológica se adapta a los requisitos cambiantes o si llegaran tarde al desarrollo de software?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	13	¿Considera usted que la Propuesta Metodológica fomenta el trabajo en equipo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	14	¿Cree usted que se logrará la entrega a tiempo de los productos desarrollados bajo los parámetros de la seguridad con la aplicación de la Propuesta Metodológica?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Nelson Agurto	15	¿Encuentra usted correcto el empleo de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	1	¿Cree usted que la Propuesta Metodológica cubre la necesidad de agregar el atributo de seguridad en las etapas del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	2	¿Considera usted que la Propuesta Metodológica cubre todos los aspectos necesarios del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	3	¿Cree usted que aplicando la Propuesta Metodológica se optimiza la seguridad en el proceso de desarrollo de software?	Fácil



V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	4	¿Cree usted que al aplicar la Propuesta Metodológica se lograría cumplir con los tiempos establecidos en la Desarrollo de un proyecto sin sacrificar la seguridad del software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	5	¿Cree usted que se podría entregar un software que satisfaga las aspiraciones del cliente y los usuarios, con la aplicación de la Propuesta Metodológica?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	6	¿Cree usted que la Propuesta Metodológica cumple con los objetivos requeridos por el Departamento de Desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	7	¿Cree usted que la Propuesta Metodológica contribuye con controles para mejorar la Desarrollo establecida por el equipo de técnico?	Estándar
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	8	¿Cree usted que es posible solucionar las falencias de seguridad de software presentadas hasta el momento, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Michael Jiménez	9	¿Considera usted que la Propuesta Metodológica mantiene una estructura lógica, organizada y estructurada para satisfacer el ciclo de vida del software?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	10	¿Considera usted que la Propuesta Metodológica se acopla ágilmente a cualquier proyecto de software que se desarrolla en el Departamento Desarrollo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	11	¿Cree usted que el esfuerzo utilizado por el personal técnico para aplicar la Propuesta Metodológica está en los parámetros normales?	Fácil

V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	12	¿Considera usted que la Propuesta Metodológica se adapta a los requisitos cambiantes o si llegarán tarde al desarrollo de software?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	13	¿Considera usted que la Propuesta Metodológica fomenta el trabajo en equipo?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	14	¿Cree usted que se logrará la entrega a tiempo de los productos desarrollados bajo los parámetros de la seguridad con la aplicación de la Propuesta Metodológica?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Michael Jiménez	15	¿Encuentra usted correcto el empleo de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	1	¿Cree usted que la Propuesta Metodológica cubre la necesidad de agregar el atributo de seguridad en las etapas del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	2	¿Considera usted que la Propuesta Metodológica cubre todos los aspectos necesarios del proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	3	¿Cree usted que aplicando la Propuesta Metodológica se optimiza la seguridad en el proceso de desarrollo de software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	4	¿Cree usted que al aplicar la Propuesta Metodológica se lograría cumplir con los tiempos establecidos en la Desarrollo de un proyecto sin sacrificar la seguridad del software?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	5	¿Cree usted que se podría entregar un software que satisfaga las aspiraciones del cliente y los usuarios, con la aplicación de la Propuesta Metodológica?	Estándar

V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	6	¿Cree usted que la Propuesta Metodológica cumple con los objetivos requeridos por el Departamento de Desarrollo?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	7	¿Cree usted que la Propuesta Metodológica contribuye con controles para mejorar la Desarrollo establecida por el equipo de técnico?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	8	¿Cree usted que es posible solucionar las falencias de seguridad de software presentadas hasta el momento, con la aplicación de la Propuesta Metodológica?	Fácil
V.D. Minimización de vulnerabilidades del software frente ataques informáticos	Entrevista 2	Ing. Com. María Paz Ludeña	9	¿Considera usted que la Propuesta Metodológica mantiene una estructura lógica, organizada y estructurada para satisfacer el ciclo de vida del software?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	10	¿Considera usted que la Propuesta Metodológica se acopla ágilmente a cualquier proyecto de software que se desarrolla en el Departamento Desarrollo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	11	¿Cree usted que el esfuerzo utilizado por el personal técnico para aplicar la Propuesta Metodológica está en los parámetros normales?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	12	¿Considera usted que la Propuesta Metodológica se adapta a los requisitos cambiantes o si llegaran tarde al desarrollo de software?	Fácil
V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	13	¿Considera usted que la Propuesta Metodológica fomenta el trabajo en equipo?	Estándar
V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	14	¿Cree usted que se logrará la entrega a tiempo de los productos desarrollados bajo los parámetros de la seguridad con la aplicación de la Propuesta Metodológica?	Fácil

V.I. Propuesta Metodológica	Entrevista 2	Ing. Com. María Paz Ludeña	15	¿Encuentra usted correcto el empleo de la Propuesta Metodológica?	Fácil
-----------------------------	--------------	----------------------------	----	---	-------