



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

MANTENIMIENTO DE LA APLICACIÓN WEB DE INFORMACIÓN PRODUCTIVA AGROPECUARIA CONAGOPARE CHIMBORAZO

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentar para optar al grado académico de:

INGENIERO DE SOFTWARE

AUTORES:

ALEX SANTIAGO CUNACHI CHACHA

ERICK FABRICIO DÁVALOS MANCERO

Riobamba – Ecuador

2022



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA SOFTWARE

MANTENIMIENTO DE LA APLICACIÓN WEB DE INFORMACIÓN PRODUCTIVA AGROPECUARIA CONAGOPARE CHIMBORAZO

Trabajo de Integración Curricular

Tipo: Proyecto Técnico

Presentar para optar al grado académico de:

INGENIERO DE SOFTWARE

AUTORES: ALEX SANTIAGO CUNACHI CHACHA

ERICK FABRICIO DÁVALOS MANCERO

DIRECTORA: Ing. Gloria de Lourdes Arcos Medina

Riobamba – Ecuador

2022

© 2022, Alex Santiago Cunachi Chacha, Erick Fabricio Dávalos Mancero

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el derecho de Autor

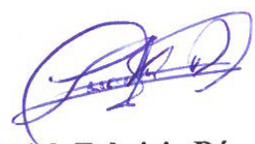
Nosotros, **Alex Santiago Cunachi Chacha** y **Erick Fabricio Dávalos Mancero**, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular, el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, viernes 09 de diciembre de 2022



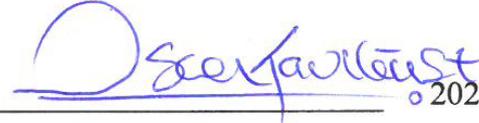
Alex Santiago Cunachi Chacha
180493852-8



Erick Fabricio Dávalos Mancero
172560986-9

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA SOFTWARE

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Proyecto Técnico, **MANTENIMIENTO DE LA APLICACIÓN WEB DE INFORMACIÓN PRODUCTIVA AGROPECUARIA CONAGOPARE CHIMBORAZO**, realizado por los señores: **Alex Santiago Cunachi Chacha y Erick Fabricio Dávalos Mancero** ha sido minuciosamente revisado por los Miembros del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIDMA	FECHA
Ing. Ivonne Elizabeth Rodríguez Flores PRESIDENTE DEL TRIBUNAL		2022-12-08
Ing. Gloria de Lourdes Arcos Media DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2022-12-08
Ing. Oscar Danilo Gavilánez Álvarez ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2022-12-08

DEDICATORIA

Este trabajo quiero dedicar con mucho cariño al ser más grandioso y extraordinario que existe Dios por haberme permitido cumplir mi objetivo día a día por ser quien con sus bendiciones me tiene con vida. A los seres más importantes de mi vida que sin ellos no hubiera cumplido esta meta, a mi familia en especial a mi abuela, padres, tíos. A mis padres Carlos y Nelly quienes me apoyaron durante toda mi formación académica y con sus consejos y apoyo pude lograrlo. A mis amigos por haberme brindado su amistad incondicionalmente con quienes compartí muchos momentos siempre los tendré en mi mente y en mi corazón.

Erick

El presente trabajo de integración curricular lo dedico principalmente a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados. A mis padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy. A mis hermanos porque me acompañaron cada día y por brindarme su apoyo moral, a lo largo de mi vida. A todas las personas que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y me compartieron sus conocimientos.

Alex

AGRADECIMIENTO

Quiero expresar mi gratitud a DIOS por haberme dado a los mejores padres quienes me dieron la vida y haberme otorgado su total comprensión, apoyo y sobre todo la fortaleza. Mi sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, institución de gran reconocimiento y prestigio por haberme brindado sabios conocimientos para mi formación académica. A la Ing. Gloria de Lourdes Arcos Media, Director de Tesis, también al Ing. Oscar Danilo Gaviláñez Álvarez Miembro de Tesis, gracias por su asesoramiento y aporte para la realización de mi investigación y paciencia para cumplir mi objetivo. A todos los maestros de las diferentes materias de la Escuela de Software, gracias por sus enseñanzas, conocimientos y por formarnos como excelentes profesionales y buenos seres humanos. A mi compañero Alex Cunachi, compañero que me ha ayudado mucho en la consecución y elaboración de este proyecto. Sin él muchas cosas me habrían resultado mucho más difíciles.

Erick

Quiero expresar mi gratitud a Dios, quien con su bendición llena siempre mi vida y a toda mi familia por estar siempre presentes. Mi profundo agradecimiento a todo el personal que componen CONAGOPARE Chimborazo por confiar en mí, abrirme las puertas y permitirme realizar todo el proceso de este presente trabajo de integración curricular dentro de su establecimiento. De igual manera mis agradecimientos a la Escuela Superior Politécnica de Chimborazo, a toda la Facultad de Informática y electrónica, a mis profesores en especial a la Ing. Gloria Arcos y el Ing. Oscar Gaviláñez quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad lo cual permitieron la culminación con éxito del presente trabajo de integración curricular. Por último, agradeciendo a mi compañero Erick Dávalos que son su apoyo se pudo lograr cumplir los objetivos propuestos.

Alex

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE ANEXOS	xi
RESUMEN	xii
SUMMARY	xiii
INTRODUCCIÓN	1

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA	3
1.1. Antecedentes.....	3
1.2. Formulación del problema.....	5
1.3. Sistematización del problema	5
1.4. Justificación teórica	5
1.5. Justificación aplicativa	8
1.6. Objetivos.....	9
1.6.1. Objetivo General.....	9
1.6.2. Objetivos Específicos	9

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS	10
2.1. Mantenimiento de software	10
2.1.1. Causas y costos del mantenimiento de software	10
2.1.2. Definiciones y terminología.....	10
2.1.3. Tipos de mantenimiento de software	11
2.1.3.1. Mantenimiento Correctivo.....	11
2.1.3.2. Mantenimiento Perfectivo.....	11
2.1.3.3. Mantenimiento Adaptativo.....	12
2.1.3.4. Mantenimiento Preventivo	12
2.2. Seguridad informática.....	14
2.2.1. Los pilares de la seguridad.....	15
2.2.1.1. Confidencialidad.....	16
2.2.1.2. Integridad.....	16
2.2.1.3. Disponibilidad	16
2.2.2. Etapas o niveles de madurez de la seguridad	16

2.3.	Vulnerabilidades emitidas por la Fundación OWASP	17
2.4.	Buenas prácticas de seguridad	19
2.5.	Metodología Mantema	21
2.5.1.	Características de la metodología Mantema	21
2.5.2.	Estructura general de la metodología	22
2.5.3.	Integración de procesos	22
2.5.4.	Participantes del proceso de mantenimiento	23
2.5.5.	Ciclo de vida del mantenimiento	23
2.5.6.	Estructura detallada de la metodología	25
2.5.6.1.	Actividades y tareas iniciales comunes	26
2.5.6.2.	Actividades y tareas del mantenimiento no planificable	29
2.5.6.3.	Actividades y tareas del mantenimiento planificable	29
2.5.6.4.	Actividades y tareas finales comunes	32
2.6.	ISO/IEC 25010	33
2.6.1.	Adecuación funcional (Funcionalidad)	34
2.7.	Herramientas y metodologías para el desarrollo y mantenimiento del software 34	
2.7.1.	JavaScript	34
2.7.2.	Node JS	35
2.7.3.	PHP	35
2.7.4.	Gestor de base de datos MYSQL	36
2.7.5.	Modelo vista controlador (MVC)	36
2.7.6.	Kali Linux	37
2.7.6.1.	Características de Kali Linux.....	37
2.8.	Pruebas de penetración	39
2.9.	Trabajos relacionados	39

CAPÍTULO III

3.	MARCO METODOLÓGICO	41
3.1.	Tipo de investigación	41
3.2.	Métodos y técnicas	41
3.3.	Lugar de estudio	42
3.4.	Determinación de la funcionalidad	43
3.5.	Determinación de la seguridad	44
3.6.	Planteamiento de la hipótesis	46
3.6.1.	Operacionalización de variables	47

3.7.	Aplicación de la metodología Mantema en el sistema de información productiva CONAGOPARE Chimborazo	47
3.7.1.	Fase 1: Identificación, clasificación y priorización del problema	48
3.7.1.1.	Actividades y tareas iniciales	48
3.7.2.	Fase 2: Análisis	50
3.7.2.1.	Actividades y tareas del mantenimiento planificable.....	50
3.7.3.	Fase 3: Diseño y desarrollo	52
3.7.3.1.	Actividad P2. Intervención y pruebas	52
3.7.3.2.	Análisis y rediseño de la base de datos	52
3.7.3.3.	Análisis de la arquitectura del sistema	53
3.7.4.	Fase 4: Finalización del proceso de mantenimiento de software	58
3.7.4.1.	Actividades y Tareas Finales Comunes	58

CAPÍTULO IV

4.	RESULTADOS Y DISCUSIÓN	60
4.1.	Resultados del indicador de funcionalidad	60
4.1.1.	Resultados obtenidos de los errores de funcionalidad en el sistema	60
4.1.2.	Resultados obtenidos de completitud funcional	62
4.1.3.	Resultados obtenidos de exactitud funcional	63
4.1.4.	Resultados obtenidos de pertinencia funcional	63
4.1.5.	Resultados generales de funcionalidad antes y después de aplicar el proceso de mantenimiento de software	64
4.2.	Resultados obtenidos del indicador de seguridad	67
4.2.1.	Análisis de resultados de seguridad antes de realizar el proceso de mantenimiento de software	67
4.2.2.	Análisis de resultados de seguridad después de realizar el proceso de mantenimiento de software	70
4.2.3.	Comparativo de resultados de seguridad antes y después de aplicar el mantenimiento de software	72
4.2.4.	Prueba chi-cuadrado (x2)	77
	CONCLUSIONES	80
	RECOMENDACIONES	82
	GLOSARIO	
	BIBLIOGRAFÍA	

ÍNDICE DE TABLAS

Tabla 1-2:	Crecimiento estadístico por años de los costos de mantenimiento de software	13
Tabla 2-2:	Buenas prácticas de seguridad en aplicaciones web.....	20
Tabla 1-3:	Métodos y técnicas por objetivos propuestos	41
Tabla 2-3:	Indicadores de funcionalidad según ISO/IEC 25010	43
Tabla 3-3:	Resumen de ataques al sistema de información productiva agropecuaria CONAGOPARE Chimborazo	45
Tabla 4-3:	Operacionalización de variables.....	47
Tabla 5-3:	Personal involucrado	49
Tabla 6-3:	Estimaciones por el método T-Shirt.....	55
Tabla 7-3:	Modelo de historia de usuario	56
Tabla 8-3:	Modelo de prueba de aceptación	57
Tabla 1-4:	Resultados de los usuarios en la parte de funcionalidad del sistema, mediante la manipulación por 30 minutos antes del mantenimiento de software.....	61
Tabla 2-4:	Ponderación de prioridad de los indicadores de funcionalidad	64
Tabla 3-4:	Resultados generales de indicadores de funcionalidad antes del mantenimiento de software	65
Tabla 4-4:	Resultados generales de indicadores de funcionalidad después del mantenimiento de software	65
Tabla 5-4:	Tipos de ataque de penetración por identificador.....	67
Tabla 6-4:	Resultados por caso de uso del porcentaje de vulnerabilidad en sql Injection antes del mantenimiento de software.....	67
Tabla 7-4:	Resultados obtenidos de los ataques antes del mantenimiento de software	68
Tabla 8-4:	Resultados de los ataques antes del mantenimiento de software mediante estadística descriptiva.....	69
Tabla 9-4:	Resultados por caso de uso del porcentaje de vulnerabilidad en sql injection después del mantenimiento de software	70
Tabla 10-4:	Resultados obtenidos de los ataques después del mantenimiento de software..	70
Tabla 11-4:	Resultados de los ataques después del mantenimiento de software mediante estadística descriptiva.....	72
Tabla 12-4:	Comparativa general de los ataques de penetración en el sistema	73
Tabla 13-4:	Resumen de resultados por indicadores CIA en el sistema de información productiva agropecuario CONAGOPARE Chimborazo	76
Tabla 14-4:	Tabla de contingencia (Confidencialidad, integridad y general)	77
Tabla 15-4:	Resultados obtenidos de la prueba del chi cuadrado	78
Tabla 16-4:	Resultados obtenidos de la prueba chi-cuadrado (Confidencialidad, integridad y general).....	78

ÍNDICE DE ILUSTRACIONES

Ilustración 1-2:	Los tres pilares de la seguridad.....	15
Ilustración 2-2:	Macroestructura del Proceso de Mantenimiento de ISO/IEC 12207	22
Ilustración 3-2:	Visión General de los Procesos en la Metodología.....	22
Ilustración 4-2:	Actividades Cíclicas en la Metodología.....	25
Ilustración 5-2:	Actividades y tareas metodología Mantema	26
Ilustración 1-3:	Diagrama de componentes modelo-vista-controlador del sistema.....	54
Ilustración 1-4:	Comparativa de los resultados de completitud funcional	63
Ilustración 2-4:	Comparativa de los resultados de errores de pertinencia funcional	64
Ilustración 3-4:	Comparativa de los resultados de ponderación por indicador	66
Ilustración 4-4:	Comparativa de los resultados de media de funcionalidad	66
Ilustración 5-4:	Porcentaje de error de los ataques realizados al sistema antes del mantenimiento de software.....	69
Ilustración 6-4:	Porcentaje de error de los ataques realizados al sistema después del mantenimiento de software.....	71
Ilustración 7-4:	Confidencialidad e integridad antes y después del mantenimiento	73
Ilustración 8-4:	Disponibilidad antes y después de aplicar buenas prácticas de seguridad en el proceso de mantenimiento de software.....	74
Ilustración 9-4:	Resultados generales antes y después de aplicar buenas prácticas de seguridad en el proceso de mantenimiento de software.....	75
Ilustración 10-4:	Comparativa de resultados por indicadores de seguridad (CIA)	76
Ilustración 11-4:	Distribución chi-cuadrado (Confidencialidad, integridad y general).....	79

ÍNDICE DE ANEXOS

Anexo A: Proceso de mantenimiento de software: Mantema

Anexo B: Valores obtenidos de las pruebas de penetración a las vulnerabilidades del software

Anexo C: Manual de buenas prácticas de seguridad

RESUMEN

El objetivo del presente Trabajo de Integración Curricular, fue realizar el proceso de mantenimiento de software en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo, para la ejecución del proyecto se usó la metodología Mantema debido a que está enfocada al mantenimiento de software, permitiendo tener una documentación organizada debido a que se sigue un proceso de actividades y tareas. Mediante un análisis, se determinó el tipo de mantenimiento a realizar, que en este proyecto fue de tipo perfectivo porque se incrementaron las funcionalidades que fueron obtenidas mediante una entrevista al cliente encargado de la aplicación, así se desarrollaron 25 funcionalidades pasando de 32 antes del mantenimiento a 57 realizadas después del mantenimiento; también fue de tipo correctivo debido a que se realizaron cambios en interfaces de usuario y bugs; además se aplicó buenas prácticas de seguridad en el sistema. La evaluación del proyecto se realizó mediante los indicadores de funcionalidad y seguridad, por lo que mediante las pruebas de penetración se obtuvieron resultados iniciales antes de realizar el mantenimiento de software con la finalidad tener datos de comparación. Se realizó el proceso de medición de funcionalidad mediante los sub-indicadores obtenidos de la ISO 25010 que son completitud, exactitud y pertinencia funcional. Por otro lado, para la medición de seguridad se utilizó los sub-indicadores de confidencialidad, integridad, disponibilidad (CIA), obtenidos de la ISO 27001. Se concluye con los resultados, para funcionalidad fueron favorables debido a que se obtuvo una disminución de los errores funcionales en un 22% para completitud y pertinencia funcional. Los resultados de seguridad tras aplicar 7 tipos de ataques de penetración a las 51 funcionalidades desarrolladas después del mantenimiento fueron favorables obteniendo una minimización de vulnerabilidad del 47,50%; se recomienda el uso del manual de buenas prácticas de seguridad.

Palabras clave: <INGENIERÍA DE SOFTWARE>, <MANTENIMIENTO DE SOFTWARE>, <SEGURIDAD INFORMÁTICA>, <PRUEBAS DE PENETRACIÓN>, <METODOLOGÍA MANTEMA>, <ISO 25010>, <ISO 27001>, <INDICADORES DE SEGURIDAD DE LA INFORMACIÓN>.



SUMMARY

The objective of this Curriculum Integration Work was to carry out the software maintenance process in the agricultural productive information system CONAGOPARE Chimborazo. We utilized the Mantema methodology for the execution of the project because it is focused on software maintenance, allowing to have an organized documentation since it follows a process of activities and tasks. We determined the type of maintenance to be performed through an analysis, it was a perfective maintenance in this project because we increased the functionalities that were obtained through an interview with the client in charge of the application, thus 25 functionalities were developed, going from 32 before maintenance to 57 after maintenance. It was also corrective because we made changes in user interfaces and bugs; in addition, we applied good security practices in the system. The evaluation of the project was carried out by means of functionality and security indicators, so we obtained initial results through pen testing before performing software maintenance in order to have comparison data. We carried out the functionality measurement process using the sub-indicators obtained from ISO 25010 standard, which are completeness, accuracy, and functional relevance. On the other hand, for the security measurement we utilized the sub-indicators of confidentiality, integrity, availability (CIA), obtained from ISO 27001 standard. We conclude that the results for functionality were favorable due to a 22% decrease in functional errors for completeness and functional relevance. The security results after applying 7 types of penetration attacks to the 51 functionalities developed after maintenance were favorable, obtaining a vulnerability minimization of 47.50%. The use of the good security practices manual is recommended.

Key words: <SOFTWARE ENGINEERING>, <SOFTWARE MAINTENANCE>, <IT SECURITY>, <PEN TESTING>, <MANTEMA METHODOLOGY>, <ISO 25010 STANDARD>, <ISO 27001 STANDARD>, <INFORMATION SECURITY INDICATORS>.

Translated by:



Lic. Carolina Campaña D. Mgs.
ID number: 1804191482
EFL Professor

INTRODUCCIÓN

La tecnología ha sido positiva y negativa en la sociedad, pero se ha vuelto indispensable para todos ya que ha ayudado a que el progreso en varios campos sea más rápido y productivo. En Ecuador el campo tecnológico ha avanzado mucho en cuanto a aplicaciones web, la gran mayoría de empresas está implementando aplicaciones web que se orientan a identificar necesidades de implementar y automatizar tecnología web para mejorar la productividad de ciertas entidades.

CONAGOPARE Chimborazo, es una instancia asociativa, con potestad de ejercer la vocería oficial ante instituciones públicas y privadas, construyendo, definiendo y fortaleciendo la gestión de las políticas públicas en función de fortalecer la autonomía y unidad de los Gobiernos parroquiales de la provincia de Chimborazo, esta institución maneja una revisión de la cantidad de productos agropecuarios que se han producido en un lapso de tiempo, en las diferentes parroquias rurales pero era tedioso, hacer este registro debido a que los encargados, tenían que viajar a las distintas parroquias para obtener la información, por este motivo se creó una aplicación en la web donde facilite el trabajo dando así la oportunidad de que cada encargado de las 45 parroquias rurales existentes en la provincia de Chimborazo suba la información correspondiente de los productos agropecuarios producidos en un determinado tiempo.

El sistema de información productiva agropecuaria CONAGOPARE Chimborazo he tenido su desarrollo, pero al haber un incremento en sus funcionalidades, seguridad y corrección de errores, tras una investigación y análisis se vio la necesidad de realizar mantenimiento de software con la finalidad de corregir fallos en interfaz de usuario, aumentar funcionalidades para cumplir con los objetivos para la cual fue desarrollada la aplicación web, integrar seguridad al sistema y realizar pruebas, para verificar su vulnerabilidad antes ataque informáticos debido a que estos permiten la sustracción de información personal e importante para la Institución.

Para el proceso de mantenimiento de software se va a realizar con las mismas herramientas propuestas o documentas, mismo lenguaje de programación PHP, se utiliza la metodología Mantema debido a que está enfocada en el mantenimiento de software, se aplica buenas prácticas de seguridad antes ataques de vulnerabilidades presentes en sistemas web documentadas según un listado publicado por la fundación OWASP empresa sin lucro dedicada a buscar vulnerabilidades en los sistemas web, documentar y publicar ante usuarios que requieran esa información importante.

El presente documento está compuesto de cuatro capítulos los cuales se describen a continuación:

Capítulo I: Diagnóstico del problema, aquí se describe el planteamiento del problema, la justificación y los objetivos a cumplirse en el trabajo de integración curricular.

Capítulo II: Fundamentos teóricos, en este capítulo se describe definiciones y conceptos de las herramientas, métodos, lenguajes y metodologías utilizadas en la aplicación del mantenimiento de software.

Capítulo III: Marco metodológico, en este capítulo contiene las actividades y tareas a realizarse en la metodología Mantema utilizada en el mantenimiento de software.

Capítulo IV: Resultados, este capítulo contiene los resultados obtenidos después de aplicar el proceso de mantenimiento de software, y de ser evaluado el sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

En este presente capítulo se presenta la problemática a la situación a resolver, su justificación aplicativa y teórica del por qué se va a resolver el problema y los objetivos a proponer para el presente trabajo de integración curricular.

1.1. Antecedentes

Desde hace muchos años de la creación del internet y avances tecnológicos, se han desarrollado un sin número de aplicaciones, programas o software que facilitan el trabajo en varios procesos que se realizan en empresas o en lugares donde brindan un servicio pero, debido a que los sistemas software tiene un ciclo de vida, en ellas está inmersa la parte de mantenimiento que es de suma importancia debido a que según la terminología ANSI-IEEE, el mantenimiento del software es: “la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno”.

CONAGOPARE Chimborazo, es una instancia asociativa, con potestad de ejercer la vocería oficial ante instituciones públicas y privadas, construyendo, definiendo y fortaleciendo la gestión de las políticas públicas en función de fortalecer la autonomía y unidad de los Gobiernos parroquiales de la provincia de Chimborazo, esta institución maneja una revisión de la cantidad de productos agropecuarios que se han producido en un lapso de tiempo, en las diferentes parroquias rurales pero era tedioso, hacer este registro debido a que los encargados, tenían que viajar a las distintas parroquias para obtener la información, por este motivo se creó una aplicación en la web donde facilite el trabajo dando así la oportunidad de que cada encargado de las 45 parroquias rurales existentes en la provincia de Chimborazo suba la información correspondiente de los productos agropecuarios producidos en un determinado tiempo, la aplicación consta de los siguientes módulos:

- **Módulo de registro o autenticación de datos:** permite al encargado de subir la información de la parroquia ingresar al sistema.
- **Módulo de información de la parroquia:** autenticado los datos permite visualizar la información de la parroquia además de la modificación eliminación y editado de la información en cuanto a los campos de las unidades productivas agropecuarias, la superficie, hectáreas y la cobertura de riego destinadas a la producción agropecuaria.

- **Módulo de producción agrícola:** Destinado para visualizar, eliminar, editar, e ingresar información del tipo de producto agrícola que se cultiva en esa parroquia además de la cantidad de hectáreas de terreno para su producción, también la cantidad producida en un lapso de tiempo y cuanto se destina para la venta en su propia provincia y en otras provincias del Ecuador.
- **Módulo de producción pecuaria:** Contiene la información pecuaria de la parroquia que tipo de animales son criados, su tipo de raza, la cantidad destinada en cuanto a la venta dentro y fuera de la provincia, y la cantidad producida total:
- **Módulo de información gubernamental:** Debido que se necesita maquinaria y equipos para la producción en este módulo se detalla la institución que provee de esos insumos la cantidad y el tipo de material que la institución brinda.
- **Módulo de estación de comercio:** Aquí se detallan la cantidad de tiendas o mercados existentes
- **Módulo de turismo:** Destinado para guardar la información de los lugares turísticos existentes de la parroquia, su nombre y tipo.

De los módulos desarrollados se pueden realizar impresiones de tipo pantalla o exportadas a Excel para ser impresiones físicas, además de poder realizar un crud de la información.

Por motivo de que los requerimientos funcionales del sistema que se han incrementado por parte del técnico de la aplicación, no fue implementado seguridad en el sistema y posee ciertos errores que necesitan ser corregidos estas causas dan como consecuencia que al existir un incremento en su funcionalidad la aplicación no cumpla por completo sus objetivos a la cual está destinada; además que por los errores existentes pueda colapsar la aplicación y no se aprecie bien la interfaz de usuario, también debido a la falta de implementación de seguridad en el sistema y como se encuentra subida en la web está expuesta a robo de información que solo la institución maneja además de los ataques como inyecciones sql, Buffer overflow (DOS) y Cross site request forgery (CSRF) (falsificación de solicitudes entre sitios) por mencionar los más comunes, por ende, se intenta resolver al problema de que “requisitos funcionales incompletos, corrección de errores e implementación de seguridad en el sistema de producción agropecuario de Conagopare Chimborazo”.

Existen algunos trabajos que se han realizado en torno a mantenimiento y seguridad de software se puede mencionar los siguientes: El desarrollo de una gestión para mantenimiento de software basándose en una norma ISO 14764:2006 (Caldas Herrera 2015). También un trabajo de investigación de mantenimiento de software con perspectiva de seguridad (Zelkowitz, 2014).

Además de la implementación mediante un estudio sobre seguridad informática (Álvarez Marañón 2004).

Estos trabajos o investigaciones en comparación con lo propuesto a realizar dan una perspectiva de cómo realizar el mantenimiento de software, además de como se ha ido implementando la variable de seguridad y así poder tener una idea más clara de lo que se va a realizar mediante las investigaciones que se realiza. Con la finalidad de continuar con las mejoras en la aplicación se realizará el mantenimiento de software de tipo correctivo y perfectivo y así seguir desarrollando los requerimientos que se ha aumentado además de la implementación de seguridad, utilizando las herramientas que ya estaban establecidas, open source: base de datos MySQL, y el entorno de desarrollo, php para código interno y angular para el desarrollo externo o interfaz de usuario.

1.2. Formulación del problema

¿De qué manera el mantenimiento de software puede mejorar las funcionalidades y seguridad en la aplicación web de información productiva agropecuaria CONAGOPARE Chimborazo?

1.3. Sistematización del problema

¿Cuál es la arquitectura del sistema de información productiva existente?

¿Cuáles son las características de la metodología Mantema que se utiliza para el mantenimiento de software?

¿De qué manera aplicar las buenas prácticas de seguridad desarrolladas por la fundación OWASP puede disminuir las vulnerabilidades del sistema?

¿Cuáles son los módulos faltantes y correcciones que se realizarán en el proceso de mantenimiento de software de la aplicación web de información productiva agropecuaria?

¿Cómo determinar la mejora del sistema en cuanto a su funcionalidad y seguridad una vez realizado el mantenimiento en el sistema de información productiva agropecuaria existente?

1.4. Justificación teórica

Muchas de las aplicaciones creadas en la actualidad son de gran utilidad en muchos campos aplicativos debido a que se automatizan sus procesos y para la sociedad industrializada y globalizada este es un gran aporte debido a que se reduce el trabajo porque ya hoy en día se maneja a manera de un sistema informático, pero no obstante las aplicaciones en muchas ocasiones no terminan por estar completas al pasar el ciclo de vida estas requieren de un mantenimiento para corregir fallos, cambios de plataforma, implementar seguridades, dar una mejor utilidad o

implementar nuevas funcionalidades que en base se vaya utilizando se dé cuenta que necesite lograr algo más.

Como todo en la vida no es igual, lo mismo ocurre en la parte de la programación, no todo programador realiza código de la misma manera debido a que existen un sinnúmero de lenguajes, frameworks, gestores de bases de datos, hojas de estilos que se pueden implementar al momento de desarrollar una aplicación que sea requerida, por ello es que al momento de realizar un mantenimiento en un software lo primero hay que entender es la metodología, lenguaje en el que se esté realizando y las herramientas utilizadas, a continuación se presentara una breve explicación de lo que se ha implementado y se seguirá implementado en este proceso de vida del software.

Se pudo identificar que se utiliza el lenguaje php, como lo indica Miguel Álvarez que a su vez lo extrajo del libro de “Desarrollo Web PHP” escrito por el Ing. Roberto Henríquez. “PHP es el acrónimo de Hipertext Preprocesor, es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.” (Cobo et al. 2005)

Así mismo como se ocupa un lenguaje del lado del servidor para su desarrollo también se necesita un complemento del lado del cliente se identificó que se utiliza el framework angular. (Robles 2017) framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitar el desarrollo de aplicaciones web SPA y además dar herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima. Además, se explica cómo funciona aplicación web SPA creada con Angular es una web de una sola página, en la cual la navegación entre secciones y páginas de la aplicación, así como la carga de datos, se realiza de manera dinámica, casi instantánea, asíncronamente haciendo llamadas al servidor (backend con un API REST) y sobre todo sin refrescar la página en ningún momento.

“MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional” (B, 2019).

Como otro punto a implementar es la seguridad informática. pero antes hay que conocer de qué se trata. (Voutssas 2010) expone en su artículo que, para poder comprender el concepto integral de la seguridad informática, es indispensable entender los diversos conceptos básicos que la rigen, ya que de otra forma no es posible establecer una base de estudio como son:

Recursos Informáticos: el equipo de cómputo y telecomunicaciones; los sistemas, programas y aplicaciones, así como los datos e información de una organización. También se les conoce como "activos informáticos"

- **Amenaza:** fuente o causa potencial de eventos o incidentes no deseados que pueden resultar en daño a los recursos informáticos de la organización.
- **Impacto:** la medida del efecto nocivo de un evento.
- **Vulnerabilidad:** característica o circunstancia de debilidad de un recurso informático la cual es susceptible de ser explotada por una amenaza.
- **Riesgo:** la probabilidad de que un evento nocivo ocurra combinado con su impacto en la organización.

El objetivo primario de la seguridad informática es el de mantener al mínimo los riesgos sobre los recursos informáticos, y garantizar así la continuidad de las operaciones de la organización al tiempo que se administra ese riesgo informático a un cierto costo aceptable.

El objetivo secundario de la seguridad informática consiste en garantizar que los documentos, registros y archivos informáticos de la organización mantengan siempre su confiabilidad total. Este concepto varía de acuerdo a distintos autores, a los contextos documentales y al tipo de organización a la que la información esté asociada. En un contexto archivístico y en donde se trata de interoperar un enfoque de seguridad informática con uno de preservación digital, se puede establecer esa confiabilidad como la unión de seis características esenciales:

- Permanencia
- Accesibilidad
- Disponibilidad
- Confidencialidad (privacidad)
- Autenticidad (integridad)
- Aceptabilidad (no repudio)

Entonces debido a que se establecieron estas herramientas se continuará el trabajo con lo predispuesto además que se mencionó su aplicación y como ventajas que pueden ayudar en tanto el lenguaje como el framework que para realizar interfaz amigable es la más recomendada debido a que consume menos recursos y es más eficiente.

1.5. Justificación aplicativa

El Mantenimiento de software que se ha propuesto para este trabajo de integración curricular permitirá la corrección de errores, incrementación de nuevas funcionalidades e implementación de seguridad en el sistema, esto permitirá que la aplicación cumpla con su objetivo por la cual ha sido creada con todos los parámetros y protección en cuanto a robo de información, mediante el mantenimiento correctivo y perfectivo se desarrollará los siguientes módulos:

- **Módulos de empresas productivas**, permite llevar un control de cuántos empleados tiene cada institución comercial en la provincia de Chimborazo.
- **Componente de movilidad información sobre estado**, mantenimiento y kilometraje de las vías existentes en la provincia de Chimborazo.
- **Modulo del administrador**, que permitirá realizar reportes de los usuarios existentes además de la gestión de los mismos
- **Módulo económico social**, permite llevar un control de construcciones de infraestructuras como canchas deportivas.
- **Módulo de gestión documental**, información personal que solo se maneja entre los Gads parroquiales de Chimborazo

Debido a que la aplicación de información productiva agropecuaria está montada en la web la implementación de seguridad proporcionará estabilidad en cuanto a la protección de la información que dicha aplicación contiene permitirá evitar ataques como inyección sql, robos y manipulación de la información que solo los técnicos de los Gads parroquiales de Chimborazo tienen acceso, para continuar con el desarrollo del resto de funcionalidades mencionadas en el anterior párrafo se seguirá utilizando las mismas herramientas propuestas como son la programación del lado del servidor que se realiza en php, para interfaz de usuario o programación del lado del cliente se utilizará el framework angular, con la base de datos se continua con MySQL y en cuanto a la metodología se procederá con SCRUM y Mantema, además que para la parte de seguridad se lo realizará mediante la aplicación de buenas prácticas emitidas por la Fundación OWASP encargada de disminuir las vulnerabilidades de sistemas software.

De acuerdo con el eje de investigación de la Escuela Superior Politécnica de Chimborazo (ESPOCH), la presente propuesta del proyecto de integración curricular se ubica en el eje de TICS en la línea de investigación de líneas transversales – tecnologías de la información y comunicación (Tic's) en el programa de ingeniería de software y seguridad de sistemas de información, en el ámbito métodos empíricos en el desarrollo y mantenimiento de productos de software, gestión de información.

1.6. Objetivos

1.6.1. Objetivo General

Realizar el mantenimiento de software de la aplicación web de información productiva agropecuaria para la institución “CONAGOPARE Chimborazo”.

1.6.2. Objetivos Específicos

- Analizar la arquitectura del sistema de información productiva agropecuaria para determinar la estructura del sistema desarrollado.
- Determinar las características de la metodología Mantema para documentar el proceso relativo al mantenimiento de software
- Aplicar buenas prácticas de seguridad desarrolladas por la fundación OWASP (Open Web Application Security Project) para minimizar vulnerabilidades del sistema información productiva.
- Desarrollar los módulos de empresas productivas, movilidad, económico social y gestión documental y corrección de errores del sistema de información productiva agropecuaria para una correcta funcionalidad del sistema.
- Determinar el nivel de funcionalidad y seguridad del sistema a través de pruebas de penetración para su evaluación.

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

En este capítulo se presenta la recopilación de antecedentes, investigaciones previas y consideraciones teóricas, contextuales o legales de los conceptos que se utilizaron para el planteamiento del problema de la investigación y que sustenta el trabajo de integración curricular.

2.1. Mantenimiento de software

2.1.1. Causas y costos del mantenimiento de software

Antes que nada, en el artículo de (Ruiz y Polo 2001) menciona sobre el ciclo de vida del software y relaciona la complejidad del proceso de producción de software se intenta abordar mediante la descomposición en diversas etapas. Esta descomposición ha recibido el nombre de ciclo de vida del software. Los diversos modelos de ciclo de vida que han sido propuestos plantean variantes a partir de las siguientes fases principales:

- Análisis y definición de requisitos.
- Especificación.
- Diseño.
- Programación (escritura del código).
- Prueba e instalación.
- Operación y mantenimiento.

Por tanto, las tareas de mantenimiento son las últimas en realizarse en el ciclo de vida clásico del software.

Y en esta investigación (Ruiz y Polo 2001) le da un gran énfasis a la parte de costos debido a que, múltiples estudios señalan que el mantenimiento es la parte más costosa del ciclo de vida del software. Estadísticamente está comprobado que el coste de mantenimiento de un producto software a lo largo de toda su vida útil supone más del doble que los costes de su desarrollo.

2.1.2. Definiciones y terminología

Se debe tener en claro lo que es o como se define un mantenimiento de software, por lo que se realizó una investigación de diferentes autores.

Según (Pressman 1998) menciona que “la fase de mantenimiento se centra en el cambio que va a asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el

entorno del software, y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente”.

Por otro lado, un estándar utilizado para el mantenimiento de software IEEE 1219 escrito en el documento de (IEEE 1993) lo define como “Modificación de un producto software después de haber sido entregado (a los usuarios o clientes) con el fin de corregir los defectos, mejorar el rendimiento u otros atributos, o adaptarlo a un cambio de entorno”.

Por lo tanto, el mantenimiento basándose en estas definiciones por autores son las modificaciones realizadas en los sistemas software con el fin de evolucionar o realizar cambios en funciones de mejoras o correcciones de errores con el objetivo de optimizar la utilidad del sistema.

2.1.3. Tipos de mantenimiento de software

Mediante la investigación de (Ruiz y Polo 2001) en el artículo publicado se puede mencionar una clasificación de los tipos de mantenimiento que se explican a continuación:

2.1.3.1. Mantenimiento Correctivo

Tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de causar un fallo. Un fallo ocurre cuando el comportamiento de un sistema es diferente del establecido en la especificación. Entre otros, los fallos en el software pueden ser de:

- Procesamiento, por ejemplo, salidas incorrectas de un programa.
- Rendimiento, por ejemplo, tiempo de respuesta demasiado alto en una búsqueda de información.
- Programación, por ejemplo, inconsistencias en el diseño de un programa.
- Documentación, por ejemplo, inconsistencias entre la funcionalidad de un programa y el manual de usuario.

2.1.3.2. Mantenimiento Perfectivo

Los cambios en la especificación, normalmente debidos a cambios en los requerimientos de un producto software, implican un nuevo tipo de mantenimiento llamado perfectivo. La casuística es muy variada. Desde algo tan simple como cambiar el formato de impresión de un informe, hasta la incorporación de un nuevo módulo funcional. Se puede definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario. Algunos autores dividen este tipo de mantenimiento en dos:

- Mantenimiento de Ampliación: orientado a la incorporación de nuevas funcionalidades.
- Mantenimiento de Eficiencia: que busca la mejora de la eficiencia de ejecución.

2.1.3.3. Mantenimiento Adaptativo

Este tipo de mantenimiento consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el cual se ejecuta. Los cambios pueden afectar a:

- Sistema operativo (cambio a uno más moderno),
- La arquitectura física del sistema informático (paso de una arquitectura de red de área local a Internet/Intranet),
- Entorno de desarrollo del software (incorporación de nuevos elementos o herramientas como ODBC).

Los cambios en el entorno software pueden ser de dos clases:

- En el entorno de los datos, por ejemplo, al dejar de trabajar con un sistema de ficheros clásico y sustituirlo por un sistema de gestión de bases de datos relacionales.
- En el entorno de los procesos, por ejemplo, migrando a una nueva plataforma de desarrollo con componentes distribuidos, Java, ActiveX, etc.

Este tipo de mantenimiento es cada vez más frecuente debido principalmente al cambio, cada vez más rápido, en los diversos aspectos de la informática: nuevas generaciones de hardware, nuevos sistemas operativos o versiones de los antiguos, y mejoras en los periféricos o en otros elementos del sistema (frente a esto, la vida útil de un sistema software puede superar fácilmente los diez años).

2.1.3.4. Mantenimiento Preventivo

Este último tipo de mantenimiento consiste en la modificación del software para mejorar las propiedades de dicho software (por ejemplo, aumentando su calidad y/o su mantenibilidad) sin alterar sus especificaciones funcionales. Algunas maneras de hacerlo son: Incluir sentencias que comprueben la validez de los datos de entrada, reestructurar los programas para mejorar su legibilidad, o incluir nuevos comentarios que faciliten la posterior comprensión del programa. En algunos casos se ha planteado el Mantenimiento para la Reutilización, consistente en modificar el software (buscando y modificando componentes para incluirlos en bibliotecas) para que sea más fácilmente reutilizable. En realidad, este tipo de mantenimiento es preventivo, especializado en mejorar la propiedad de reusabilidad del software.

Una vez realizado la investigación acerca de los tipos de mantenimiento, se tiene una idea más clara a que está enfocada este tema de trabajo de integración curricular y se puede determinar que el tipo de mantenimiento que se va a realizar es correctivo porque se va a corregir errores que se presentan en la aplicación además que también es perfectivo debido a que se va a realizar un aumento en sus funcionalidades. La tendencia es creciente con el paso del tiempo como se muestra en la **Tabla 1-2**.

Tabla 1-2: Crecimiento estadístico por años de los costos de mantenimiento de software

Referencia	Fechas	% de mantenimiento
(Pressman, 1993)	años 70	35% - 40%
(Lientz y Swanson, 1980)	1976	60%
(Pigoski, 1997)	1980-1984	55%
(Rock-Evans y Hales, 1990)	1987	67%
(Schach, 1990)	1987	67%
(Frazer, 1992)	1990	80%

Fuente: (Ruiz y Polo 2001)

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Se puede concluir que los costos de mantenimiento de software se han elevado, pues debido a que existen sistemas más complejos la tecnología evoluciona, se crean nuevas herramientas, lenguajes de codificación.

Además, menciona que son varias las causas de que en la mayoría de las organizaciones actuales se requiera mucho trabajo de mantenimiento:

- Una gran cantidad del software que existe actualmente ha sido desarrollada hace más de 10 años. Aunque estos programas fuesen creados utilizando las mejores técnicas de diseño y codificación existentes en su momento (la mayoría no lo fueron).
- Estos programas han sufrido una o varias migraciones a nuevas plataformas o sistemas operativos.
- Todos estos cambios se realizaron sin tener en cuenta la arquitectura general del sistema (no se aplicaron técnicas de ingeniería inversa o reingeniería).
- Otra causa directa de los grandes costes del MS es que el coste relativo de reparar un defecto aumenta considerablemente en las últimas etapas del ciclo de vida del software, de forma que la relación entre el coste de detectar y reparar un defecto en la fase de análisis de requisitos y en la fase de mantenimiento es de 1 a 100 respectivamente.

Algunas de las razones por las que es menos costoso detectar y corregir un error durante las etapas iniciales del ciclo de vida que durante las etapas últimas son:

- Es más fácil cambiar la documentación (por ejemplo, los documentos de especificación o de diseño) que modificar el código.
- Es más fácil encontrar un defecto durante la fase en la cual se ha introducido el defecto que tratar de detectar y corregir los efectos provocados por el defecto en una fase posterior.
- La causa de un defecto puede esconderse en la inexistencia o falta de actualización de los documentos de especificación o diseño.

2.2. Seguridad informática

Según (Aguilera 2011) , se puede definir a la seguridad informática como la disciplina encargada de plantear y diseñar las normas, procedimientos, métodos y técnicas con el fin de obtener que un sistema de información sea seguro, confiable y sobre todo que tenga disponibilidad.

Mientras que (Aguirre 2006) afirma que la seguridad informática puede definirse como el conjunto de métodos y de varias herramientas para proteger el principal activo de una organización como lo es la información o los sistemas ante una eventual amenaza que se pueda suscitar.

La principal tarea de la seguridad informática es la de minimizar los riesgos, en este caso provienen de muchas partes, puede ser de la entrada de datos, del medio que transporta la información, del hardware que es usado para transmitir y recibir, los mismos usuarios y hasta por los mismos protocolos que se están implementando, pero siempre la tarea principal es minimizar los riesgos para obtener mejor y mayor seguridad.

En la publicación de (Romero Castro et al. 2018) lo que debe contemplar la seguridad se puede clasificar en tres partes como son los siguientes:

- **Los usuarios** son considerados como el eslabón más débil de la cadena, ya que a las personas es imposible de controlar, un usuario puede un día cometer un error y olvidar algo o tener un accidente y este suceso puede echar a perder el trabajo de mucho tiempo, en muchos casos el sistema y la información deben de protegerse del mismo usuario.
- **La información** Se considera como el oro de la seguridad informática ya que es lo que se desea proteger y lo que tiene que estar a salvo, en otras palabras, se le dice que es el principal activo.

- **La infraestructura** Esté puede ser uno de los medios más controlados, pero eso no implica que sea el que corre menos riesgos, siempre dependerá de los procesos que se manejan. Se deben de considerar problemas complejos, como los de un acceso no permitido, robo de identidad, hasta los daños más comunes, por ejemplo, robo del equipo, inundaciones, incendios o cualquier otro desastre natural que puede tener el material físico del sistema de la organización.

2.2.1. Los pilares de la seguridad

Según (Romero Castro et al. 2018) Los datos son valores, números, medidas, textos, documentos en bruto, la información es el valor de esos datos, es lo que aporta conocimiento. Los manuales de procedimientos, los datos de los empleados, de los proveedores y clientes de la empresa, la base de datos de facturación son datos estructurados de tal forma que se convierten en información, que aportan valor como compañía. Los pilares de la seguridad de la información se fundamentan en esa necesidad que todos tienen de obtener la información, de su importancia, integridad y disponibilidad de la información para sacarle el máximo rendimiento con el mínimo riesgo. La **Ilustración 1-2** muestra los principales pilares de la seguridad de la información.



Ilustración 1-2: Los tres pilares de la seguridad

Fuente: (Romero Castro et al. 2018)

Según (Romero Castro et al. 2018) detalla lo más importante de cada pilar con el fin de poder verificar definiciones y su importancia en la implementación de la seguridad en los sistemas informáticos.

2.2.1.1. Confidencialidad

Consiste en asegurar que sólo el personal autorizado accede a la información que le corresponde, de este modo cada sistema automático o individuo solo podrá usar los recursos que necesita para ejercer sus tareas, para garantizar la confidencialidad se recurre principalmente a tres recursos:

- Autenticación de usuarios
- Gestión de privilegios
- Cifrado de información

2.2.1.2. Integridad

Es el segundo pilar de la seguridad, consiste en asegurarse de que la información no se pierde ni se ve comprometida voluntaria e involuntariamente, el hecho de trabajar con información errónea puede ser tan nocivo para las actividades como perder la información, de hecho, si la manipulación de la información es lo suficientemente sutil puede causar que se arrastre una cadena de errores acumulativos y que sucesivamente se tome decisiones equivocadas.

2.2.1.3. Disponibilidad

Para poder considerar que se dispone de una seguridad mínima en lo que a la información respecta, se tiene a la disponibilidad, de nada sirve que solo el usuario acceda a la información y que sea incorruptible, si el acceso a la misma es tedioso o imposible, la información para resultar útil y valiosa debe estar disponible para quien la necesita, se debe implementar las medidas necesarias para que tanto la información como los servicios estén disponibles.

2.2.2. Etapas o niveles de madurez de la seguridad

Siguiendo al autor (Gómez Vieites 2006) se puede distinguir las siguientes etapas o niveles de madurez en la gestión de la seguridad de la información en una organización como son:

- **Implantación** de medidas básicas de seguridad por sentido común: consiste en la implantación en la organización de medidas primarias de seguridad, tales como: copias de seguridad o control de acceso a los recursos informáticos.
- **Adaptación** a los requisitos del marco legal y de las exigencias de los clientes: la empresa comienza a tomar conciencia de la importancia de la seguridad y cumple con la normativa legal de aplicación, como la protección de datos de carácter personal.

- **Gestión integral de la seguridad de la información:** en esta etapa, se comienza a definir políticas de seguridad, planes y procedimientos para el análisis y gestión de los riesgos en la seguridad de la información.
- **Certificación de la gestión de la seguridad de la información:** en esta etapa, se pretende certificarse en materia de seguridad de la información, para lo que se suele recurrir a un proceso de certificación basado en estándares como la ISO.

2.3. Vulnerabilidades emitidas por la Fundación OWASP

Según (Rivera 2019) menciona todo lo que se necesita saber sobre OWASP (Open Web Application Security Project), es un proyecto sin ánimo de lucro a nivel mundial que busca mejorar la seguridad del software en general. Para esto, la organización se ha provisto de una serie de herramientas y documentos que explican cuáles son las brechas de seguridad más comunes en cualquier sistema de información. Sobra decir, que todos los materiales de OWASP están disponibles de manera libre (gratuita) para su libre consulta y uso.

De acuerdo a la información publicada por la fundación OWASP a continuación se presenta un listado de las 10 vulnerabilidades que afectan a los sistemas web:

- A1: Injection
- A2: Broken Authentication
- A3: Sensitive Data Exposure
- A4-XML External Entities (XXE)
- A5-Broken Access Control
- A6-Security Misconfiguration
- A7-Cross-Site Scripting (XSS)
- A8-Insecure Deserialization
- A9: Using Known Vulnerable Components
- A10-Insufficient Logging & Monitoring

Mediante la publicación de (Fernández 2020) se obtiene definiciones de cada vulnerabilidad emitida por la fundación OWASP y se enlista a continuación:

- **Injection (Inyección)** Básicamente, consiste en inserción de código malicioso a un intérprete que forma parte de comando o query. Se debe tener en cuenta que los comandos o queries forman parte de las bases de datos. El código malicioso es capaz de ejecutar muchas acciones

no deseadas como el acceso a datos sin autorización. Los ataques de tipo inyección SQL o protocolos de acceso a directorio como LDAP son ampliamente utilizados.

- **Broken Authentication (Perdida de autenticación)** Existen ocasiones en que las funciones de autenticación y gestión de sesiones se implementan de manera incorrecta. Los errores son una pasarela para que los cibercriminales realicen acciones maliciosas como la vulneración de contraseñas, claves, tokens de sesiones etc. También pueden explotar otros fallos que les permiten asumir identidades de otros usuarios de forma temporal o permanente.
- **Sensitive Data Exposure (Exposición de datos confidenciales)**. El propósito original de las APIs es facilitar el acceso y la integración de los datos e incluso funcionalidades de los programas con otros. Sin embargo, no todas ellas tienen cuidado respecto a la protección de datos sensibles, especialmente si se trata de datos bancarios, financieros, de salud y otros. La exposición de estos datos es una ayuda extra a los cibercriminales para ejecutar otros ataques, como robo de identidad o fraudes con tarjetas de crédito. He ahí la importancia de la encriptación de los datos siendo almacenados, en tránsito o en el mejor de los casos, de extremo a extremo (End-to-End).
- **XML External Entities (XXE) (Entidades externas XML)**. El propósito de esta vulnerabilidad es que el cibercriminal consiga enviar un documento de formato XML manipulado de manera tal que, mediante un «parser XML» (un puente) comience a exponer información confidencial de las aplicaciones que son víctimas. Estos parsers se encuentran disponibles en aplicaciones compatibles con éstos. En definitiva, también pueden causar consumo excesivo de recursos, ejecución de comandos maliciosos y mucho más. Sin embargo, ¿qué es un documento XML? Es un documento de marcado que sirve para representar información estructurada en la web. En consecuencia, las aplicaciones y dispositivos son capaces de almacenar esa información, además de manipularla, visualizarla e imprimirla.
- **Broken Access Control (Pérdida de control de acceso)**. Los cibercriminales pueden explotar fallos y vulnerabilidades de las políticas de restricciones a los usuarios. Se debe tener en cuenta que estas restricciones determinan lo que puede hacer un usuario de acuerdo a los roles y permisos que tienen. Al explotar fallos y vulnerabilidades, se obtienen permisos y funcionalidades no autorizadas. Como la visualización de archivos sensibles, modificación de datos de otros usuarios, cambiar los permisos y roles de otros usuarios y mucho más.
- **Security Misconfiguration (Configuración incorrecta de seguridad)**. Acostumbra a darse a raíz de configuración por defecto inseguras, configuración ad hoc incompletas,

almacenamiento abierto en la nube, cabeceras HTTP mal configuración y mensajes de error que contienen información sensible. Aunque los sistemas operativos, frameworks, librerías y aplicaciones estén configuradas con seguridad, éstos también deberían actualizarse con parches de seguridad en forma periódica.

- **Cross-Site Scripting (XSS) (Secuencias de comandos entre sitios).** Gracias a la falta de validaciones en las aplicaciones en general, esto puede ocurrir y tener consecuencias devastadoras. XSS permite a los cibercriminales la ejecución de scripts en lenguajes como JavaScript en las sesiones de los usuarios para alterar la seguridad de los inicios de sesión en los sitios o redireccionar a dichos usuarios a sitios web maliciosos, aunque escriban una URL válida o legítima.
- **Insecure Deserialization (Deserialización insegura).** La deserialización es la reconstrucción de un objeto en base a la información recuperada del mismo. Existen unas reglas propias de cierta convención de programación que dicen que se recuperará la información que provenga de objetos serializables. Esto tiene como gran riesgo, la ejecución remota de código malicioso de toda clase, además de realizar ataques, repeticiones de ataques anteriores, inyecciones y escalada de privilegios.
- **Using Known Vulnerable Components (Uso de componentes vulnerables conocidos).** Existen muchísimas librerías, frameworks y otros módulos de software que cuentan con los mismos privilegios que las propias aplicaciones. Si se explotan componentes vulnerables, entonces existe un gran riesgo de que se den pérdidas de datos o que cibercriminales se apropien de éstos. Por eso es muy importante tener conocimiento respecto a las medidas de seguridad de las APIs que se pueda implementar, por ejemplo, para evitar abrir la puerta a un gran número de vulneraciones y ataques.
- **Insufficient Logging & Monitoring (Registro y monitoreo insuficientes)** Este par de acciones de carácter preventivo, de no estar presente en la implementación de las aplicaciones, no solo permite que los cibercriminales ejecuten los ataques si no también, puedan llegar a un nivel alto de persistencia. Es posible que puedan llegar a la instancia de destruir los datos, cosa que puede generar grandes secuelas en el entorno donde se encuentra.

2.4. Buenas prácticas de seguridad

El presente trabajo de integración curricular abarca la seguridad informática implementando buenas prácticas, pero para ello se necesita comprender su definición o para qué está predispuesto

una buena práctica, Según (Montoro 2020) menciona que las buenas prácticas hace referencia a toda experiencia que se guía por principios, objetivos y procedimientos apropiados o pautas aconsejables que se adecúan a una determinada perspectiva normativa o a un parámetro consensuado, así como también toda experiencia que ha arrojado resultados positivos, demostrando su eficacia y utilidad en un contexto concreto. Por tanto, el objetivo de las buenas prácticas son pasos o procedimientos que ayudan a la mejora de la calidad de los productos, bienes y servicios de una organización y así obtener un mejor y óptimo desempeño en el proceso final.

Anteriormente se mencionó que mediante la Fundación OWASP se determinó 10 principales vulnerabilidades en los sistemas informáticos, por lo que a continuación se detallan buenas prácticas ante los ataques o vulnerabilidades más comunes presentes en aplicaciones web **Tabla 2-2**.

Tabla 2-2: Buenas prácticas de seguridad en aplicaciones web

Vulnerabilidades	Buenas Prácticas
SQL Injection	<ul style="list-style-type: none"> • Usar PreparedStatrments en vez de Statements en la recuperación de consultas SQL. • Validar entradas en el lado del cliente con el archivo JavaScript validaciones.
Cross site scripting (XSS)	<ul style="list-style-type: none"> • Validación de entradas con la filtración de caracteres especiales en parámetros de formato establecido. • Filtración de etiquetas HTML, mediante el archivo JavaScript validaciones.
Buffer overflow (DOS)	<ul style="list-style-type: none"> • Filtrar la longitud del valor del parámetro para evitar la saturación de memoria del servidor.
Directorio transversal	<ul style="list-style-type: none"> • Validar entradas en el lado del cliente mediante el archivo JavaScript validaciones.
Cross site request forgery (CSR)	<ul style="list-style-type: none"> • Mitigación basada en token • Implementar la librería CSRF Guard de OWASP Foundation (OWASP Foundation, 2018).
Intercepción criptográfica	<ul style="list-style-type: none"> • Encriptar las claves de autenticación • Instalar certificados SSL, con esto se encriptará el canal de comunicación cuando exista tráfico en la red.
Password mismanagement	<ul style="list-style-type: none"> • Combina de forma aleatoria letras en mayúscula, letras en minúscula, números y símbolos • Usar al menos 15 caracteres

Fuente: (Ordoñez Granizo y Chimbo Ortiz 2019)

Realizado por: Alex Cunachi y Erick Dávalos, 2022

2.5. Metodología Mantema

Según (Balseca Rodríguez y Caiza Chacón 2014) la metodología Mantema muestra la visión del proceso de mantenimiento desde el mayor nivel de abstracción, en el que probablemente no interesa el contenido de las instrucciones ni los campos de los archivos, aunque sí las mejores técnicas para entenderlos y modificarlos. Desde este punto de vista, el proceso puede verse como el conjunto de todas las operaciones que es necesario realizar sobre el software para implementar las modificaciones solicitadas. Sin embargo, para dotar a este conjunto de operaciones de una base metodológica, es preciso definir con anterioridad el propio proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se ejecute conforme una instancia de un proceso de mantenimiento predefinido.

2.5.1. Características de la metodología Mantema

Las características de la metodología Mantema se basan en una serie de técnicas y métricas, cuya utilización se recomienda como apoyo al proceso de mantenimiento antes descrito, en esta sección se describirán estas métricas y técnicas.

- Su principal característica es la que guía el proceso de documentación y mantenimiento de software de las aplicaciones a ser modificadas.
- Usa una técnica que identifica, las características de los proyectos de mantenimiento que más riesgos producen. (Identificación y estimación de riesgos)
- Tiene un modelo económico del proyecto de mantenimiento, que permite calcular la cantidad de recursos que deben dedicarse a un periodo a servir peticiones de mantenimiento no planificable. (Estimación de recursos para el mantenimiento no planificable)
- Contiene una serie de métricas de mantenibilidad de esquemas conceptuales y lógicos de base de datos.
- También contiene un conjunto de métricas de tamaño y complejidad de sistemas orientadas a objeto.
- Por último, un conjunto de métricas para controlar el proceso de mantenimiento.

2.5.2. Estructura general de la metodología

Según (Balseca Rodríguez y Caiza Chacón 2014b) el estudio detallado del proceso de mantenimiento de ISO/IEC 12207 permitió la identificación de la macroestructura mostrada en la **Ilustración 2-2**. Tal estructura fue elegida como punto de partida para esta metodología.

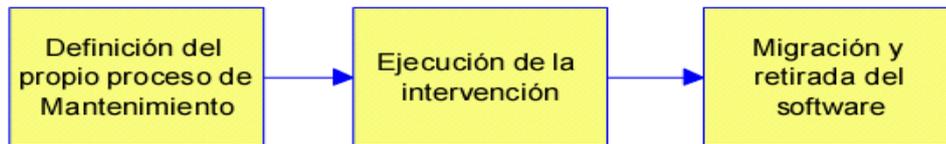


Ilustración 2-2: Macroestructura del Proceso de Mantenimiento de ISO/IEC 12207

2.5.3. Integración de procesos

Según (Cuevas Callisaya 2006) la metodología pretende la definición de un único proceso de aquéllos que componen el ciclo de vida software el mantenimiento. Durante éste se realizan trabajos de análisis y diseño, programación, pruebas, documentación, etc. Se pretende que esta metodología constituya una guía completa de mantenimiento, por lo que integra y describe, un único proceso software, las actividades de los procesos habituales del ciclo de vida software. Esquemáticamente, el resultado de tal integración puede verse de la forma expresada en la **Ilustración 3-2**, en la que se observan, como procesos satélites al mantenimiento y otros procesos del ciclo de vida cuya integración no se ha llevado a cabo pero que, sin embargo, son utilizados por el proceso de mantenimiento estableciendo interfaces en algunas tareas:

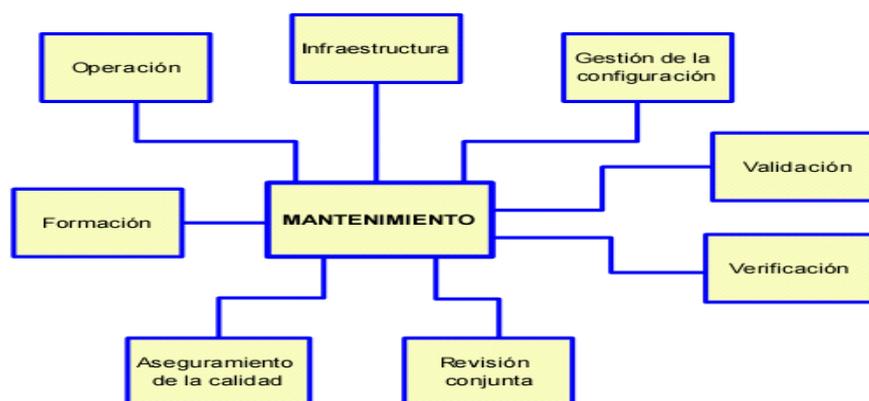


Ilustración 3-2: Visión General de los Procesos en la Metodología

2.5.4. Participantes del proceso de mantenimiento

Mediante (Piattini Velthuis et al. 2018) Mantema distingue tres organizaciones que pueden participar en el proceso de mantenimiento y estas a su vez constan de perfiles dentro de ella. Dependiendo de la situación de cada organización pueden coincidir varias en una sola. Lo mismo puede ocurrir con los perfiles. A continuación, se definen las organizaciones y sus participantes dentro del proceso de mantenimiento descritos en Mantema.

a) **Cliente:** Es la organización propietaria del software, por tanto, la que recibe el servicio de mantenimiento, dentro de esta organización se encuentra los siguientes perfiles:

- **Solicitante:** es quien presenta las solicitudes de modificación. Establece los requerimientos necesarios para su implementación y los entrega a la organización de mantenimiento.
- **Organización del Sistema:** es el departamento que conoce el sistema que será mantenido.
- **Atención a Usuarios:** es el departamento que presta asistencia a los usuarios.

b) **Organización de Mantenimiento:** es la organización que realiza el servicio de mantenimiento y consta de los siguientes perfiles:

- **Gestor de peticiones:** acepta o rechaza las peticiones modificación y decide el tipo de mantenimiento que debe aplicarse.
- **Planificador:** planifica la cola de peticiones de modificación aceptadas.
- **Equipo de Mantenimiento:** es el grupo de personas que implementa la solicitud de modificación.
- **Responsable de Mantenimiento:** prepara la etapa de mantenimiento, y establece las normas y procedimientos necesarios para llevar a cabo la metodología de mantenimiento usada.

c) **Usuario:** es la Organización que utiliza el software objeto del mantenimiento.

2.5.5. Ciclo de vida del mantenimiento

Según (Ruiz y Polo 2001) el proceso de mantenimiento descrito en Mantema comienza con la solicitud de prestación de servicios de mantenimiento de una organización a otro se continua con los siguientes pasos:

- Una vez que la organización cliente se pone en contacto con la organización suministradora del servicio de mantenimiento, ésta realiza un estudio inicial del software cuyo mantenimiento

se solicita y del hardware sobre el que éste se ejecuta. Tras este estudio, se pretende que la organización que presta el servicio conozca las características del sistema de información que se va a mantener, de manera que, si procede, pueda ir preparando presupuestos, contratos de prestación de servicios, etc.

- Cuando la organización cliente acepta la propuesta de mantenimiento de la organización suministradora del servicio, se realiza una planificación del proceso a la medida de la organización cliente. Estas acciones se llevan a cabo en el conjunto de Actividades y tareas iniciales. Tras esta segunda actividad (Planificación del proceso), la responsabilidad de la ejecución de las intervenciones de mantenimiento sobre el software convenido en la actividad primera (Estudio inicial) recae ya sobre la organización de mantenimiento, que estará preparada para recibir peticiones de modificación, clasificarlas de acuerdo a uno de los tipos de mantenimiento definidos en Mantema y servir las conforme a las actividades (y a sus correspondientes tareas) enunciadas en los nodos centrales de la **Ilustración 4-2**.
- Ya servida completamente la petición (lo cual incluye el paso a producción del software modificado), la organización de mantenimiento llega al nodo final, en el que, para cada petición de modificación, debe ejecutar algunas de las Actividades y tareas finales, comunes también a todos los tipos de mantenimiento. En particular, debe realizarse la actualización de la base de datos histórica para almacenar toda la información relevante de la petición, registrar la intervención y, en caso necesario, proceder a la retirada del software. La actividad denominada Fin de la externalización es realizada cuando ha existido una relación contractual de prestación de servicios de mantenimiento entre organizaciones diferentes que llega a su punto final o, más genéricamente, cuando va a cambiar la organización suministradora del servicio de mantenimiento.

Se observan dos conjuntos de actividades claramente diferentes en la **Ilustración 4-2**. El primero, formado por las actividades que, por lo general, son realizadas solamente una vez, tanto al principio como al final del proceso de mantenimiento (estudio inicial y planificación del proceso, en las actividades y tareas iniciales, y fin de la externalización, en las actividades y tareas finales). El segundo, compuesto de un conjunto de actividades repetitivo, que se ejecuta una vez para cada petición de modificación recibida.

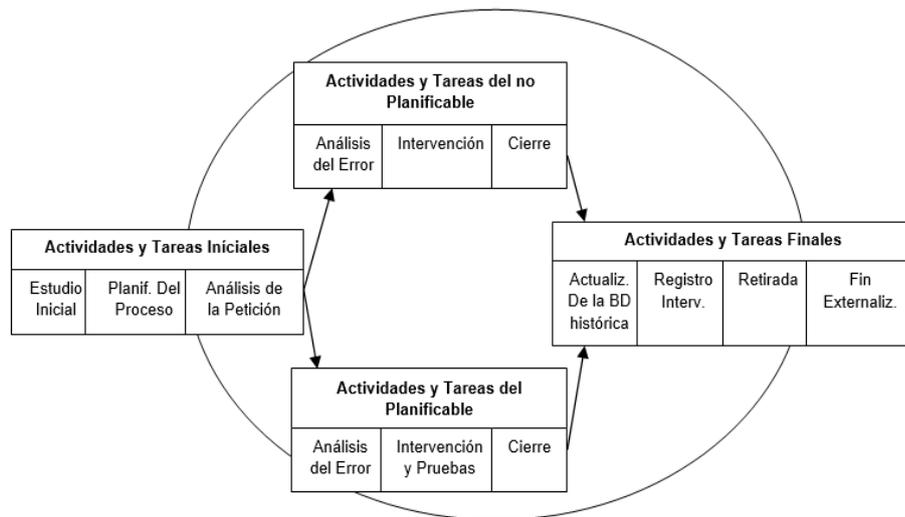


Ilustración 4-2: Actividades Cíclicas en la Metodología

2.5.6. Estructura detallada de la metodología

En esta sección se muestra en resumen las actividades y tareas iniciales, propias de cada tipo de mantenimiento y finales del tipo de metodología para el desarrollo del mantenimiento de software. Se presenta a continuación detalladamente en la **Ilustración 5-2**

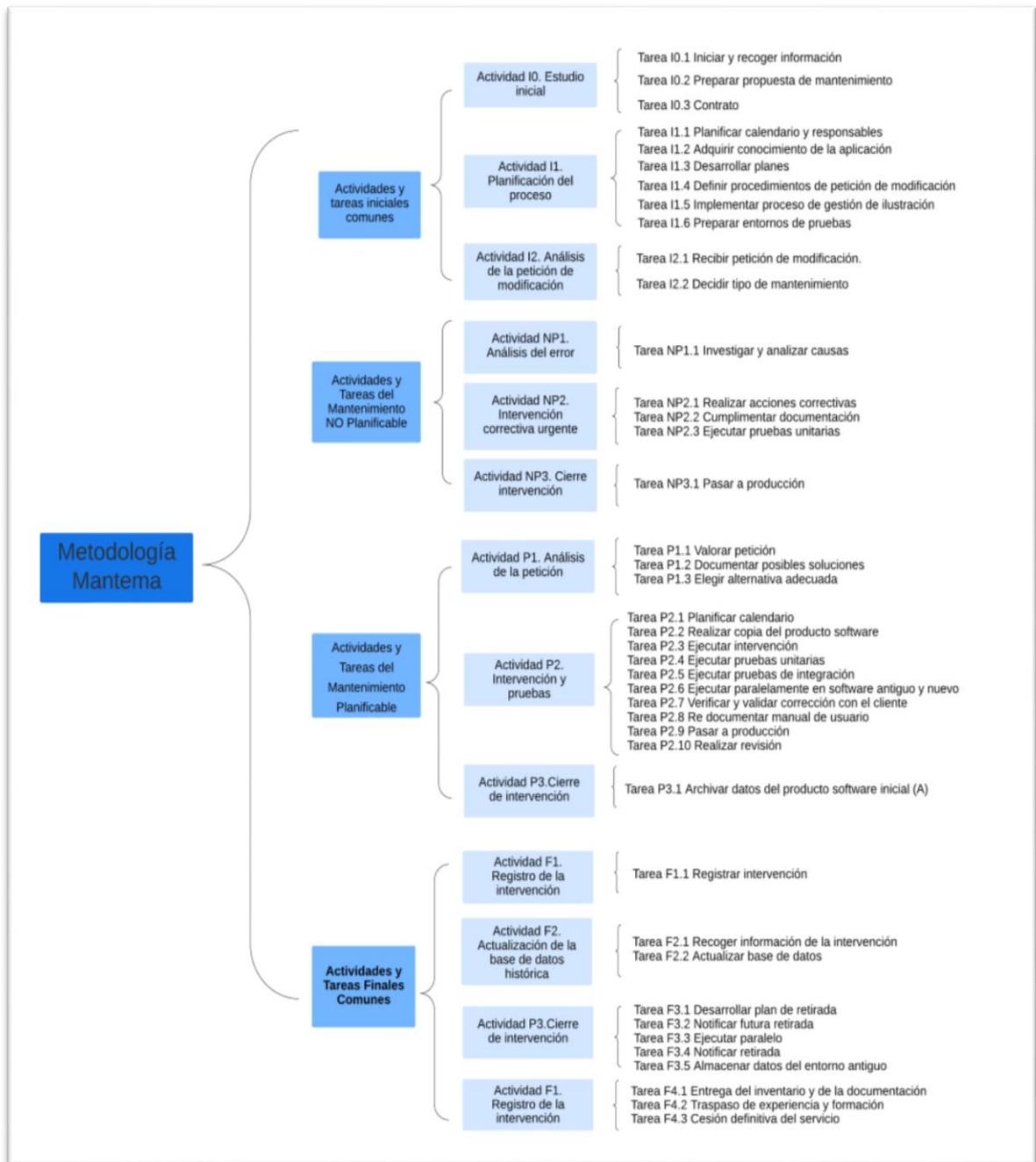


Ilustración 5-2: Actividades y tareas metodología Mantema

Realizado por: Alex Cunachi y Erick Dávalos, 2022

2.5.6.1. Actividades y tareas iniciales comunes

En la guía realizada por el grupo Alarcos y mediante la investigación de (Ruiz y Polo 2001) enlista las tareas y actividades del proceso de mantenimiento utilizando la metodología Mantema y se muestra a continuación:

a) **Actividad I0. Estudio inicial**

- **Tarea I0.1 Iniciar y recoger información.** En esta tarea, como respuesta a una solicitud de prestación del servicio de mantenimiento, el equipo de mantenimiento y la organización del sistema rellena un cuestionario inicial en el que detalla ciertos aspectos del software que se deberá mantener (sistema operativo, lenguaje o lenguajes de programación, número de programas y cualesquiera otros datos que puedan resultar de interés para la organización de mantenimiento), según una muestra tomada del producto software.
- **Tarea I0.2 Preparar propuesta de mantenimiento.** De los datos recogidos en la Tarea I0.1, el responsable de mantenimiento crea una propuesta de mantenimiento del software. En la preparación de la propuesta (a la que, en su caso, acompañará una propuesta económica), puede utilizarse la técnica de identificación de riesgos.
- **Tarea I0.3 Contrato.** Se redacta y se firma el contrato de mantenimiento entre la organización del sistema y el responsable de mantenimiento. Esta tarea puede omitirse si ambas organizaciones son la misma, aunque su ejecución sea obligatoria cuando exista externalización.

b) **Actividad I1. Planificación del proceso**

- **Tarea I1.1 Planificar calendario y responsables.** Se genera un calendario de reuniones y se enumeran los interlocutores válidos y los responsables de ambas partes. Las reuniones planificadas pueden servir como hitos de comprobación de resultados.
- **Tarea I1.2 Adquirir conocimiento de la aplicación.** En esta tarea, el Equipo de mantenimiento debe estudiar la documentación existente del software objeto de la prestación del servicio, el código de los programas, referencias cruzadas, entrevistarse con los usuarios, observar cómo trabaja la organización de mantenimiento actual, etc. Durante esta tarea, el encargado de resolver las peticiones de modificación es el propio cliente (o la organización que tenga en ese momento contratada); sin embargo, el equipo de mantenimiento observa su trabajo para conocer mejor el software objeto de la prestación del servicio. esta tarea puede durar entre uno y dos meses, durante los cuales el equipo de mantenimiento no habrá probablemente modificado una sola línea de código. Al finalizar esta tarea, el equipo de mantenimiento debe entregar al cliente un informe acerca del estado de su software, de manera que, entre otras cosas, se permite verificar que el equipo de mantenimiento ha adquirido un conocimiento adecuado del software que deberá mantener. Por otro lado, las

circunstancias pueden hacer que esta tarea no sea ejecutada total o parcialmente (el software no ha sido mantenido con anterioridad, por ejemplo).

- **Tarea I1.3 Desarrollar planes.** El equipo de mantenimiento desarrolla los planes de mantenimiento y construye el resumen técnico del aplicativo, una vez adquirido el conocimiento. Debe inventariarse el sistema de información que se va a mantener, para cada aplicación y base de datos.
- **Tarea I1.4 Definir procedimientos de petición de modificación.** El equipo de mantenimiento genera modelos de documentos para la presentación de cada petición de modificación. Así mismo, establece, junto al cliente, los procedimientos de petición de modificación, en los que indica quién será el receptor de la petición, quién es el encargado de su estudio, etc.
- **Tarea I1.5 Implementar proceso de gestión de ilustración.** Esta tarea establece una interfaz con el proceso de gestión de ilustración existente en la organización para gestionar las modificaciones que se realicen sobre el sistema existente.
- **Tarea I1.6 Preparar entornos de pruebas.** El equipo de mantenimiento prepara copias del entorno software de producción para implementar sobre ellas las intervenciones de mantenimiento.

c) **Actividad I2. Análisis de la petición de modificación**

- **Tarea I2.1 Recibir petición de modificación.** El solicitante o, en caso de estar autorizado para ello, el usuario, entrega una petición de modificación que es recibida y registrada por el gestor de peticiones, que le asigna un identificador único.
- **Tarea I2.2 Decidir tipo de mantenimiento.** A partir de la petición de modificación recibida y registrada en la Tarea I2.1, el gestor de peticiones decide o bien rechazar la petición, o bien aceptarla y decidir el tipo de mantenimiento que debe aplicarse. En caso de elegir la primera opción, deben justificarse las razones. En caso de aceptación, la petición de modificación es entregada al planificador que la distribuye y encauza al tipo de mantenimiento que corresponda.

2.5.6.2. Actividades y tareas del mantenimiento no planificable

a) Actividad NP1. Análisis del error

- **Tarea NP1.1 Investigar y analizar causas.** El equipo de mantenimiento analiza la petición de modificación, verifica el problema (quizás con la colaboración del usuario) y lo reproduce, y estudia diferentes alternativas para implementar la modificación. Además, debe construir una lista de los elementos software a corregir (módulos, rutinas, documentos, etc.).

b) Actividad NP2. Intervención correctiva urgente

- **Tarea NP2.1 Realizar acciones correctivas.** El equipo de mantenimiento ejecuta las acciones necesarias para corregir el problema detectado. Deben identificarse las rutinas y bases de datos afectadas por la intervención.
- **Tarea NP2.2 Cumplimentar documentación.** El equipo de mantenimiento debe documentar los cambios realizados en el documento de acciones correctivas realizadas.
- **Tarea NP2.3 Ejecutar pruebas unitarias.** El equipo de mantenimiento debe comprobar la corrección de todos los cambios realizados. Estas pruebas se documentarán en el documento de pruebas unitarias realizadas.

c) Actividad NP3. Cierre intervención

- **Tarea NP3.1 Pasar a producción.** El equipo de mantenimiento pasa al entorno de producción el software corregido para su utilización por parte de los usuarios. Se establece aquí una interfaz con el proceso de gestión de la configuración para garantizar la corrección de la tarea.

2.5.6.3. Actividades y tareas del mantenimiento planificable

Puesto que no todas las actividades del mantenimiento planificable son aplicables a todos los tipos de mantenimiento que se ha englobado bajo esta denominación (correctivo no urgente, perfectivo, preventivo y adaptativo) ya que conservan algunas diferencias, se indica las siguientes claves los tipos de mantenimiento a los que cada tarea es aplicable. Las claves serán CP para el mantenimiento correctivo no urgente y perfectivo. Y para el mantenimiento adaptativo y P para el preventivo.

a) **Actividad P1. Análisis de la petición**

- **Tarea P1.1 Valorar petición (CP/P).** La Petición de modificación que fue recibida en la tarea I2.1 del conjunto de actividades y tareas iniciales comunes y distribuida en la Tarea I2.2- está ya en posesión del equipo de mantenimiento que va a llevar a cabo la intervención. En esta tarea, se valoran los costes y esfuerzos necesarios para servir la petición. Igualmente, se valoran la disponibilidad de recursos y de calendario para estimar los plazos de la intervención.
- **Tarea P1.2 Documentar posibles soluciones (CP/P).** A partir del producto software en explotación y de la petición de modificación:
 - Si se trata de CP (correctivo no urgente o perfectivo), se documenta la causa del error y se indican las posibles alternativas de implementación en el documento de diagnóstico del error y posibles soluciones dejando pendiente de rellenar la alternativa de implementación elegida.
 - Si se trata de P (preventivo), se produce una lista de elementos software y propiedades mejorables.
- **Tarea P1.3 Elegir alternativa adecuada (CP).** El equipo de mantenimiento completa el documento de diagnóstico del error y posibles soluciones indicando la alternativa de corrección elegida. Para la elección de la alternativa pueden realizarse estimaciones de datos de proyectos anteriores para optimizar la dedicación de recursos.

b) **Actividad P2. Intervención y pruebas**

- **Tarea P2.1 Planificar calendario (A).** El equipo de mantenimiento realiza una planificación del calendario de la intervención en el que debe detallar cuándo se acometerán las diferentes fases de la adaptación.
- **Tarea P2.2 Realizar copia del producto software (A).** Antes de iniciar la intervención adaptativa, el equipo de mantenimiento realiza una copia del producto software y de todas las bases de datos, ficheros, etc. relacionados.
- **Tarea P2.3 Ejecutar intervención (CP/P/A).** El equipo de mantenimiento debe ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada.

- **Tarea P2.4 Ejecutar pruebas unitarias (CP/P/A).** El equipo de mantenimiento realiza las pruebas unitarias sobre el producto software intervenido. Debe comprobarse que la petición de modificación queda servida. una vez finalizadas, se genera el documento de pruebas unitarias realizadas.
- **Tarea P2.5 Ejecutar pruebas de integración (CP/P/A).** El Equipo de mantenimiento debe comprobar que los diferentes elementos software funcionan correctamente de forma conjunta.
- **Tarea P2.6 Ejecutar paralelamente en software antiguo y nuevo (CP/P/A).** El equipo de mantenimiento ejecuta acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso. Pueden aplicarse pruebas de no regresión, de manera que se repiten casos de prueba anteriores a la intervención. En el caso del mantenimiento adaptativo se reconstruye, si es preciso, la documentación técnica del sistema.
- **Tarea P2.7 Verificar y validar corrección con el cliente (CP/P/A).** El equipo de mantenimiento y la Organización del sistema (o el perfil en que se haya delegado) se reúnen para comprobar que el producto intervenido funciona correctamente.
- **Tarea P2.8 Re documentar manual de usuario (CP/P/A).** El equipo de mantenimiento debe re documentar el manual de usuario, si es que ha cambiado el modo de operación del software.
- **Tarea P2.9 Pasar a producción (CP/P/A).** El equipo de mantenimiento instala el producto software modificado en el entorno de trabajo real. En el caso del mantenimiento adaptativo (y tal vez en algún caso aislado de otro tipo de mantenimiento), puede establecerse una interfaz con el proceso de infraestructura por si es preciso cambiar o agregar máquinas, etc.
- **Tarea P2.10 Realizar revisión (CP/A).** El Equipo de mantenimiento comprueba que el producto modificado funciona correctamente en el entorno de trabajo real.

c) Actividad P3. Cierre de intervención

- **Tarea P3.1 Archivar datos del producto software inicial (A).** El Equipo de mantenimiento realiza una copia de seguridad de todo lo relacionado con el producto software que se ha adaptado (programas, datos, documentación, etc.).

2.5.6.4. Actividades y tareas finales comunes

a) Actividad F1. Registro de la intervención

- **Tarea F1.1 Registrar intervención**

La intervención queda registrada, según los procedimientos de la organización.

b) Actividad F2. Actualización de la base de datos histórica

- **Tarea F2.1 Recoger información de la intervención.** Si las métricas indicadas para cada tarea no se hubieran ido recogiendo y almacenando durante la intervención, tal recolección debería ser hecha en esta tarea.

- **Tarea F2.2 Actualizar base de datos**

Los datos recogidos en la tarea anterior son incorporados a la base de datos histórica.

c) Actividad F3. Retirada

- **Tarea F3.1 Desarrollar plan de retirada.** El equipo de mantenimiento redacta un documento en el que describe cuándo se llevará a cabo la retirada del software.

- **Tarea F3.2 Notificar futura retirada.** El equipo de mantenimiento notifica al cliente el momento en el que se ejecutará la retirada.

- **Tarea F3.3 Ejecutar paralelo.** El usuario, con el visto bueno del cliente y bajo la supervisión del equipo de mantenimiento, realiza operaciones reales sobre el software que se va a retirar y el software nuevo (si es que aquél va a ser sustituido).

- **Tarea F3.4 Notificar retirada.** Se notifica la inminencia de la retirada.

- **Tarea F3.5 Almacenar datos del entorno antiguo.** Los datos del entorno antiguo son almacenados.

d) Actividad F4. Fin de la externalización

- **Tarea F4.1 Entrega del inventario y de la documentación.** Dependiendo de lo que se pactara en el contrato, posiblemente la organización de mantenimiento se encuentre obligada

a entregar al cliente todos los productos software generados y modificados durante el periodo en que ha sido responsable del mantenimiento.

- **Tarea F4.2 Traspaso de experiencia y formación.** Esta es la tarea inversa a la I1.2 (Adquirir conocimiento de la aplicación), durante la cual el equipo de mantenimiento aprendió las características del sistema observando el modo de trabajo de la organización de mantenimiento que el cliente tuviera a la sazón. En este caso, es la organización de mantenimiento la que debe formar al nuevo personal de mantenimiento. En el último periodo de esta tarea (variable según el contrato), el mantenimiento se realiza con equipos mixtos.
- **Tarea F4.3 Cesión definitiva del servicio.** La organización de mantenimiento deja de prestar sus servicios al cliente con carácter definitivo.

2.6. ISO/IEC 25010

Según (López Martínez 2017) representa el modelo de calidad y es la base a partir de la cual se establece el sistema para la evaluación de la calidad del producto. Este modelo describe una serie de características de calidad que se van a tener en cuenta para evaluar las propiedades de un producto software. El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran a continuación:

- **Adecuación funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.
- **Eficiencia de desempeño:** Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.
- **Compatibilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.
- **Usabilidad:** Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.
- **Fiabilidad:** Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.
- **Seguridad:** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
- **Mantenibilidad:** Representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

- **Portabilidad:** Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

Se concluye que en el presente trabajo de integración curricular se lo realizó mediante la característica de adecuación funcional emitida por la ISO/IEC 25010

2.6.1. Adecuación funcional (Funcionalidad)

Consiste sencillamente en que el programa haga lo que se le pide, que lo haga bien y que haga solamente eso, nada que no se pida.

Según el estándar (ISO/IEC 25000:2014 2014), la adecuación funcional: “Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas”.

Esta misma norma divide esta característica en las siguientes sub-características o indicadores:

- **Compleitud funcional.** - “Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados”. Es decir, el sistema hace lo que se pide que haga, todas funciones que el cliente/usuario pide.
- **Corrección o exactitud funcional.** - “Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido”. Lo que significa que las funcionalidades del sistema tienen el resultado esperado.
- **Pertinencia funcional.** - “Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados”. Esta es la capacidad del sistema para ejecutar únicamente las funciones necesarias para que el sistema realice las tareas y objetivos que han sido especificados por el usuario y no otros.

2.7. Herramientas y metodologías para el desarrollo y mantenimiento del software

2.7.1. JavaScript

Según (Eguiluz 2008) Lenguaje de scripting más difundido y utilizado para el desarrollo web. Fue desarrollado por Netscape para intentar dar solución a la problemática de las aplicaciones webs que cada vez eran más complejas y las velocidades de navegación (en aquel entonces) muy lentas. Este lenguaje, que se ejecuta en la computadora del cliente, permitió reducir el número de llamados al servidor, por ejemplo, cuando se necesitaban validar datos de un formulario. Hoy en día es utilizado extensamente para mejorar la experiencia del usuario a través de la incorporación

de componentes personalizados (como calendarios para seleccionar fechas o galería de imágenes), efectos y transiciones, actualizaciones de contenido sin recargar la página, entre otras características.

2.7.2. Node JS

El master en Node JS, (Pasquali 2013), nos dice que es un entorno de ejecución para JavaScript que permite interpretar dicho lenguaje fuera de las páginas web. La elección de este lenguaje se debe principalmente a su naturaleza asíncrona y orientada a eventos, lo cual permite construir aplicaciones en red escalables. Además, JavaScript es uno de los lenguajes más difundidos y conocidos en el mundo, lo cual ayudó a facilitar su adopción y contribuir a su popularidad. Uno de los principales usos que se le da es para funcionar como un servidor web. Fue desarrollado pensando principalmente en la escalabilidad y performance para permitir el procesamiento de miles de peticiones concurrentes. Posee un mecanismo de respuesta asíncrono, lo que significa que cada vez que se tiene que efectuar una operación de lectura o escritura (por ejemplo, a un archivo en disco o en una base de datos) el programa no tiene que esperar a que la operación termine, sino que prosigue con el resto de la petición o con otras peticiones y cuando la operación iniciada finaliza entonces se retoma con el proceso original y lo continúa.

2.7.3. PHP

“Lenguaje de programación con una sintaxis similar al lenguaje C, el cual es interpretado por un servidor web Apache y genera código HTML dinámico” (Cobo et al. 2005).

Entre sus ventajas se puede mencionar:

- Es de libre distribución;
- Es de código abierto, no se requiere la compra de licencias;
- Extensa comunidad de desarrolladores;
- Amplia disponibilidad de librerías de distribución gratuitas para múltiples propósitos;
- Soporte a distintos motores de base de datos, como MySQL, PostgreSQL, Oracle, MSSQL Server, entre otras;
- Fácil de aprender;
- Orientado a objetos;
- Gran variedad de frameworks de desarrollo

2.7.4. Gestor de base de datos MYSQL

Sobre este tema (Silberschatz, Korth y Sudarshan. 2002) escriben lo siguiente, un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Entre las principales ventajas del uso de bases de datos el autor enumera las siguientes:

- Centraliza y disponibilidad la información para toda persona que la requiera y tenga acceso;
- Eliminación de información redundante(duplicada) e inconsistente;
- Permite acceso concurrente por varios usuarios;
- Restricciones de seguridad a nivel de tablas, datos y operaciones sobre los mismos.

Además (Silberschatz, Korth y Sudarshan. 2002) menciona que para la implementación de aplicaciones web algunos de los DBMS más populares y utilizados son MySQL, MS SQL Server y Oracle. La decisión sobre cuál usar muchas veces se realiza teniendo en cuenta su afinidad y facilidad de integración con el lenguaje de programación elegido. MySQL suele ser utilizado para proyectos que involucran el uso de tecnologías libres, por lo que su uso va de la mano con el lenguaje PHP y los servidores Apache. MS SQL Server, al ser una tecnología de Microsoft suele utilizarse en conjunto con ASP.NET. Oracle suele utilizarse con la tecnología JSP de Java.

2.7.5. Modelo vista controlador (MVC)

Consiste en organizar los componentes de un sistema de software agrupándolos según su función. Un patrón arquitectónico que implementa esta separación en capas es MVC (Model View Controller).

Según (Moquillaza Henríquez, Vega Huerta y Guerra Grados 2010) dichas capas tienen las siguientes responsabilidades:

- **Modelo:** contiene los componentes que representan y gestionan datos relacionados con la aplicación o el negocio.
- **Vista:** es la capa de presentación formada por las interfaces. Muestran el estado actual del modelo de datos y ofrecen mecanismos de interacción.

- **Controlador:** formado por componentes que reciben las órdenes del usuario y gestionan la aplicación de la lógica de negocio sobre el modelo de datos.

También (Moquillaza Henríquez, Vega Huerta y Guerra Grados 2010) dice que la implementación de este tipo de patrones facilita el entendimiento por parte del equipo de desarrollo y facilita la incorporación de nuevos miembros, y además posibilita el crecimiento de la aplicación en su conjunto, es decir, provee escalabilidad. Esta es una característica imprescindible en un sistema CMS. Al integrar una solución de este tipo se espera que pueda encargarse de parte o de la totalidad de la capa del modelo, para dejar la lógica (controlador) y la presentación (vista) en manos de la aplicación a la que se integre.

2.7.6. Kali Linux

Para realizar las pruebas de penetración en el sistema de información productiva agropecuario se tiene la necesidad de ocupar software de testeo, para lo cual se decidió utilizar las herramientas que proporciona el S.O. de Kali Linux, a continuación, se muestra lo referente a este sistema operativo.

En la página oficial de (Kali.org 2022) describe todo lo referente a Kali Linux como que es una distribución basada en GNU/Linux Debian, orientado a auditorías de seguridad y pruebas de penetración avanzadas. Kali Linux contiene cientos de herramientas, las cuales están destinadas hacia varias tareas en seguridad de la información, como pruebas de penetración, investigación en seguridad, forense de computadoras, e ingeniería inversa. Kali Linux ha sido desarrollado, fundado y mantenido por Offensive Security, una compañía de entrenamiento en seguridad de la información. Kali Linux fue publicado en 13 de marzo del año 2013, como una reconstrucción completa de BackTrack Linux, adhiriéndose completamente con los estándares del desarrollo de Debian.

2.7.6.1. Características de Kali Linux

Por otra parte en la investigación realizada por (Caballero Quezada 2021) menciona algunas de las características que son de importancia en Kali Linux para ser una buena herramienta para pruebas de penetración en sistemas web las cuales son.

Según (Caballero Quezada 2021) Kali Linux es una completa reconstrucción de BackTrack Linux, y se adhiere completamente a los estándares de desarrollo de Debian. Se ha puesto en funcionamiento toda una nueva infraestructura, todas las herramientas han sido revisadas y empaquetadas, y se utiliza ahora Git para el VCS detalladas a continuación:

- **Incluye más de 600 herramientas para pruebas de penetración:** Después de revisar cada herramienta incluida en BackTrack, se eliminaron un gran número de herramientas, las cuales ya sea simplemente no funcionaban o duplicaban lo proporcionado por otras herramientas de funcionalidades similares.
- **Es Libre y siempre lo será:** Kali Linux como BackTrack, es completamente libre de cargo, y siempre lo será. Nunca se pagará por Kali Linux.
- **Árbol Git Open Source:** Se está comprometido con el módulo para el desarrollo de fuente abierta, y el árbol de desarrollo está disponible para todos lo vean. Todo el código fuente incluido en Kali Linux, está disponible para cualquiera quien requiera modificar o reconstruir los paquetes para satisfacer necesidades específicas.
- **Cumplimiento con FHS:** Kali Linux se adhiere al Estándar para la Jerarquía de Sistema de Archivos (Filesystem Hierarchy Standard), permitiendo a los usuarios de Linux fácilmente localizar binarios, archivos de soporte, librerías, etc.
- **Amplio soporte para dispositivos inalámbricos:** Un tema delicado con las distribuciones Linux es el soporte para las interfaces inalámbricas. Se ha construido Kali Linux para soportar tantos dispositivos inalámbricos como sea posible, permitiendo la ejecución apropiada de una amplia diversidad de hardware, haciéndolo compatible con numerosos dispositivos USB entre otros.
- **Kernel personalizado, con parches para inyección:** Como profesionales en pruebas de penetración, el equipo de desarrollo frecuentemente necesita realizar evaluaciones inalámbricas, por lo tanto, se han incluido los últimos parches para realizar inyección.
- **Es desarrollado en un entorno seguro:** El equipo de Kali Linux está constituido de un pequeño grupo de individuos, quienes son los únicos confiables para enviar paquetes e interactuar con los repositorios, todo lo cual se hace utilizando múltiples protocolos de seguridad.
- **Soporta múltiples lenguajes:** Aunque las herramientas para pruebas de penetración tienden a ser escritas en inglés, se ha asegurado Kali Linux incluya un verdadero soporte multilinguaje, permitiendo a más usuarios operarlo en su lenguaje nativo, y localizar las herramientas necesarias para su trabajo.
- **Completamente personalizable:** Se entiende no todos pueden estar de acuerdo con las decisiones hechas, por lo cual se ha facilitado tanto como sea posible; para los usuarios más aventureros; la personalización de Kali Linux, incluyendo el kernel.

Por lo tanto, Kali Linux está específicamente diseñado para las necesidades de los profesionales en pruebas de penetración, y toda la documentación asume un conocimiento previo, y familiaridad con el sistema operativo Linux en general.

2.8. Pruebas de penetración

En la investigación realizada por (Santos 2015) menciona que cuando existe incertidumbre acerca de la eficacia de los distintos mecanismos de seguridad como los controles de los firewalls, sistemas de detección de intrusiones, monitorización de integridad de archivos, etc., no son tan eficaces, lo mejor es realizar una prueba de penetración completa. Una de las técnicas más comunes para asegurarse del nivel de efectividad, es una prueba de penetración, que es un análisis de vulnerabilidad para localizar las flaquezas individuales del sistema.

Después de aplicar las buenas prácticas de seguridad en la aplicación web se pasa por un proceso de pruebas para buscar obtener resultados favorables al realizar al mantenimiento de software por tanto se debe conocer qué tipo de pruebas existen en este caso se lo realiza con pruebas de penetración.

Según (Lagos Flores 2018) menciona los tipos de pruebas de penetración existente que son:

- **Pruebas de caja blanca:** Se recibe información acerca de los sistemas tales como direcciones IP, servicios, claves de acceso, diseño de la infraestructura, permite validar parches de seguridad en servicios o servidores, realizar una prueba exhaustiva del código fuente, el cliente proporciona accesos a los dispositivos de interés para llevar a cabo las actividades.
- **Pruebas de caja gris:** El cliente brinda sólo algunos datos de información relevantes que podrían ser de ayuda al consultor a realizar la prueba, otros tantos es responsabilidad del consultor detectarlos y reportarlos.
- **Pruebas de caja negra:** En este tipo de pruebas el consultor no cuenta con información alguna para realizar la prueba, tratando de simular el acceso de un atacante externo.

Una vez realizado la investigación sobre las pruebas de penetración existentes para aplicaciones software, se optó por realizar las pruebas de este trabajo de integración curricular mediante la prueba de la caja gris debido a que el cliente en este caso el técnico de la aplicación proporcionó cierta información sobre el código, pero mediante las pruebas, se detectará vulnerabilidades de la misma.

2.9. Trabajos relacionados

Para el proceso de mantenimiento de software que se implementó en este presente trabajo de integración curricular se dio la necesidad de realizar investigación sobre trabajos relacionados que

involucren mantenimiento de software, aplicación de estándares de medición de seguridad y factibilidad, además de buenas prácticas de seguridad las cuales se mencionan a continuación.

En la tesis de (Balseca Rodríguez y Caiza Chacón 2014a) en su trabajo de titulación de “Elaboración de una guía de procedimientos para el proceso de mantenimiento de software” menciona que el proyecto de titulación que se presenta contiene una guía de procedimientos para el proceso de mantenimiento de Software, que permite a los desarrolladores seguir un conjunto de pasos para realizar el mantenimiento de sistemas software, preservando la calidad del producto y haciendo que durante este proceso el sistema no sufra un deterioro de sus características.

Según (López Martínez 2017) para la medición de funcionalidad en su trabajo de fin de grado menciona que hoy en día la calidad es una propiedad fundamental en cualquier ámbito comercial y tecnológico. Si ya de por sí la calidad es un concepto difícil de cuantificar, la calidad software es aún más difícil al ser un producto intangible y relativamente nuevo. Hasta hace poco esta calidad se medía exclusivamente a través de la calidad de los procesos del ciclo de vida (ISO/IEC 15504 e ISO/IEC 12207), pero a partir de 2005 surge el estándar ISO/IEC 25000 que permite la medición de la calidad del producto software en sí mismo.

Además(Ordoñez Granizo y Chimbo Ortiz 2019) mencionan en su trabajo de titulación que En el presente trabajo de titulación, se aplicó buenas prácticas de seguridad en el mantenimiento del sistema de Estafetas de la Dirección de Desarrollo en la Escuela Superior Politécnica de Chimborazo para influir en el nivel de vulnerabilidad; en primera instancia se procedió con la evaluación de la seguridad del sistema mediante la técnica de Pentesting, de acuerdo con los indicadores CIA (Confidencialidad, integridad y disponibilidad) definidos por el estándar ISO 27001. Al verificar la necesidad de un sistema informático seguro, se procedió con el mantenimiento del mismo mediante el uso de la metodología ágil SCRUM combinada con el estándar IEEE 1219 realizando el correspondiente análisis preliminar, planificación y mantenimiento.

Finalmente (Caldas Herrera 2015) realizó contiene el plan de mantenimiento de software basado en la norma ISO/IEC 14764 para el Departamento de Sistemas de la Universidad Politécnica Salesiana. La aplicación del mismo es de suma importancia en la arquitectura del ciclo de vida del software, puesto que proporcionará la guía para realizar el mantenimiento a las aplicaciones desarrolladas y no contratar a empresas externas. En una primera parte se describe la problemática, los objetivos, el alcance, y la revisión del estado del arte en basado en un estudio de artículos y libros.

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se incluye el tipo o tipos de investigación, las técnicas, los instrumentos, las metodologías y estándares que serán utilizados para llevar a cabo la solución al problema planteado en el presente trabajo de integración curricular.

3.1. Tipo de investigación

El trabajo de titulación corresponde a un tipo de investigación aplicada debido a que son basados en los conocimientos existentes, derivados de la investigación y/o experiencia práctica, también se enfatiza en el desarrollo de un sistema web a través de herramienta tecnológicas producciones de nuevos materiales, productos o dispositivos, al establecimiento de nuevos procesos, sistemas y servicios o a la mejora de los existentes.

También de tipo descriptiva por que se enfoca en las características externas del objeto de estudio, además que la descripción puede referirse a: personas, hechos, procesos y relaciones que pueden realizarse en un tiempo y en lugar determinado, un trabajo descriptivo generalmente utiliza medios estadísticos para presentar una situación determinada.

3.2. Métodos y técnicas

De acuerdo con los objetivos específicos planteados en el trabajo de integración curricular en la **Tabla 1-3**, se detalla los diferentes métodos y técnicas que permite dar cumplimiento de los mismos. “Los métodos y técnicas de investigación son fundamentales e indispensables para el desarrollo de un estudio de cualquier índole.

Tabla 1-3: Métodos y técnicas por objetivos propuestos

Objetivos	Métodos	Técnicas	Fuentes
Analizar la arquitectura del sistema de información productiva agropecuaria para determinar la estructura del sistema desarrollado.	Método analítico Método deductivo	<ul style="list-style-type: none">• Observación• Revisión de documentación	<ul style="list-style-type: none">• Libros• Artículos• Manuales de programación• Ficha técnica

Objetivos	Métodos	Técnicas	Fuentes
Determinar las características de la metodología Mantema para documentar el proceso relativo al mantenimiento de software	Método analítico Método deductivo	<ul style="list-style-type: none"> • Revisión de documentación 	<ul style="list-style-type: none"> • Libros • Artículos • Manuales de programación
Aplicar buenas prácticas de seguridad desarrolladas por la fundación OWASP (Open Web Application Security Project) para minimizar vulnerabilidades del sistema información productiva.	Método analítico Método deductivo	<ul style="list-style-type: none"> • Observación • Revisión de documentación 	<ul style="list-style-type: none"> • Revistas • Libros • Artículos científicos • Ficha técnica
Desarrollar los módulos de empresas productivas, movilidad, económico social y gestión documental y corrección de errores del sistema de información productiva agropecuaria para una correcta funcionalidad del sistema.	Metodología Mantema	<ul style="list-style-type: none"> • Fases de la metodología Mantema 	<ul style="list-style-type: none"> • Artículos científicos
Determinar el nivel de funcionalidad y seguridad del sistema a través de pruebas de penetración para su evaluación.	Método experimental Método estadístico	<ul style="list-style-type: none"> • Pruebas de penetración • Estándar ISO/IEC 27001, 25010 • Paper 	<ul style="list-style-type: none"> • Métricas de evaluación de funcionalidad y seguridad • Software de testeo

Realizado por: Alex Cunachi y Erick Dávalos, 2022

3.3. Lugar de estudio

El mantenimiento de software se va a realizar en el sistema de información productiva agropecuaria perteneciente a CONAGOPARE Chimborazo que es una instancia asociativa, con potestad de ejercer la vocería oficial ante instituciones públicas y privadas, construyendo, definiendo y fortaleciendo la gestión de las políticas públicas en función de fortalecer la autonomía y unidad de los Gobiernos parroquiales de la provincia de Chimborazo.

3.4. Determinación de la funcionalidad

Para determinación funcionalidad en este presente trabajo de integración curricular se tomó como población los técnicos de las parroquias de Chimborazo que en este caso son 45, por lo tanto, se realizó por muestreo intencional con un número de 11 integrantes debido a cuestiones de distancia entre las parroquias y la Institución de CONAGOPARE Chimborazo que se encuentra ubicada en la ciudad de Riobamba.

Para obtener los resultados mediante los cuales se realizan las comparativas y determinar si existe una mejora en cuanto al sistema antiguamente desarrollado y el sistema una vez aplicado el proceso de mantenimiento de software se sigue los siguientes pasos para determinar valores de funcionalidad:

- Para medir la funcionalidad primero se requirió que los 11 integrantes ocupen tanto el sistema antes de realizar el mantenimiento de software como después de haberlo hecho por un tiempo determinado de 30 minutos.
- A continuación, se realizó una tabla una matriz en donde está determinada las funcionalidades del sistema desarrollado y los técnicos que pusieron a prueba el sistema mediante la manipulación durante un período de 30 minutos, y si existía algo que no funcionaba como se espera se marca con una X la casilla de la funcionalidad donde tiene un error.
- Luego de llenar la matriz se determina el número de errores totales que tuvo el sistema, después de haber utilizado por el tiempo se realizó la prueba.
- Una vez se obtiene los resultados por los técnicos se procede a la determinación de la funcionalidad mediante los parámetros, fórmulas e indicadores que se obtiene de la **ISO/IEC 25010** como se muestra en la **Tabla 2-3**.

Tabla 2-3: Indicadores de funcionalidad según ISO/IEC 25010

Indicador	Métrica	Fórmula
Compleitud Funcional	Compleitud de la implementación funcional	$X=A/B$ A=Número de funciones que están incorrectas o que no fueron implementadas B= Numero de funciones establecidas en la especificación de requisitos
Exactitud Funcional	Exactitud	$X=A/B$ A=Número de elementos de datos implementados con el estándar especifica de exactitud B=Número total de elementos de datos implementados

Indicador	Métrica	Fórmula
	Precisión Computacional	X=A/T A=Número de cálculos inexactos encontrados T=Tiempo de operación
Pertinencia Funcional	Idoneidad Funcional	X=A/B A=Número de funciones que se detectan problemas en evaluación B=Numero de funciones evaluadas

Realizado por: Alex Cunachi y Erick Dávalos, 2022

3.5. Determinación de la seguridad

La población en el presente trabajo de integración curricular para determinar seguridad se lo realiza mediante, de los 10 ataques más frecuentes en aplicaciones web según la fundación OWASP se tomó un total de 7 ataques para pruebas de penetración en las 57 funcionalidades que corresponden al sistema de CONAGOPARE, a cada ataque se asignan las todas las funcionalidades posibles, obteniendo así, una población de 80 ataques a realizar en el sistema de información productiva CONAGOPARE Chimborazo de las cuales se puede observar en la **Tabla 3-3**.

Para determinar el tamaño de la muestra en el presente trabajo de integración curricular se obtuvo de acuerdo a la fórmula de estadística detallada a continuación.

$$n = \frac{z^2(p * q)}{e^2 + \frac{[z^2(p * q)]}{N}}$$

$$n = \frac{1.96^2(0.5 * 0.5)}{0.05^2 + \frac{[1.96^2(0.5 * 0.5)]}{121}}$$

$$n = \frac{0,9604}{0,0025 + \frac{0,9604}{80}}$$

$$n = \frac{0,9604}{0,0025 + 0,012005}$$

$$n = \frac{0,9604}{0,014505} = 66$$

Donde:

n = tamaño de la muestra
z = nivel de confianza
p = porción de la población deseada (éxito)
q = porción de la población deseada (fracaso)
e = margen de error
N = tamaño de la población

Donde:

n = tamaño de la muestra
z = 95%, entonces, 95% entre 2 = 47.5% entre 100 = 0.475 = Z = 1.96
p = 0.5
q = 0.5
e = 5% = 0.05
N = 80

Aplicando la ecuación y realizando los cálculos pertinentes se obtiene un total de 66 ataques de penetración al sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

Tabla 3-3: Resumen de ataques al sistema de información productiva agropecuaria
CONAGOPARE Chimborazo

Indicador	Tipo de ataque	Funcionalidad
Integridad Disponibilidad	A1.1 DIRECTORY TRAVERSAL	<ol style="list-style-type: none"> 1. ingresar información de la parroquia 2. modificar información de la parroquia 3. ingresar producto agrícola 4. modificar producto agrícola 5. ingresar producto pecuario 6. modificar producto pecuario 7. ingresar infraestructura gubernamental 8. obtener infraestructura gubernamental 9. ingresar estación de comercio 10. modificar estación de comercio 11. ingresar sitio turístico 12. modificar sitio turístico
Confidencialidad Integridad Disponibilidad	A1.2 CROSS-SITE REQUEST FORGERY	<ol style="list-style-type: none"> 1. ingresar información de la parroquia 2. modificar información de la parroquia 3. ingresar producto agrícola 4. modificar producto agrícola 5. ingresar producto pecuario 6. modificar producto pecuario 7. ingresar infraestructura gubernamental 8. obtener infraestructura gubernamental 9. ingresar estación de comercio 10. modificar estación de comercio 11. ingresar sitio turístico 12. modificar sitio turístico
Confidencialidad Disponibilidad	A2.1 UNENCRYPTED COMMUNICATION	<ol style="list-style-type: none"> 1. Autenticación local
Confidencialidad Integridad	A3.1 SQL INJECTION	<ol style="list-style-type: none"> 1. seleccionar información de la parroquia 2. obtener información de las parroquias 3. seleccionar producto agrícola 4. obtener productos agrícolas 5. seleccionar producto pecuario 6. obtener productos pecuarios 7. seleccionar infraestructura gubernamental 8. obtener infraestructura gubernamental 9. seleccionar estación de comercio 10. obtener estaciones de comercio 11. seleccionar sitio turístico 12. obtener sitios turísticos

Indicador	Tipo de ataque	Funcionalidad
Confidencialidad Integridad Disponibilidad	A3.2 XSS	<ol style="list-style-type: none"> 1. ingresar información de la parroquia 2. modificar información de la parroquia 3. ingresar producto agrícola 4. modificar producto agrícola 5. ingresar producto pecuario 6. modificar producto pecuario 7. ingresar infraestructura gubernamental 8. obtener infraestructura gubernamental 9. ingresar estación de comercio 10. modificar estación de comercio 11. ingresar sitio turístico
Confidencialidad Integridad	A4.1 PASSWORD MISMANAGEMENT	<ol style="list-style-type: none"> 1. Autenticación local
Confidencialidad Integridad	A4.2 PRIVILEGE ESCALATION	<ol style="list-style-type: none"> 1. ingresar información de la parroquia 2. modificar información de la parroquia 3. eliminar información de la parroquia 4. seleccionar información de la parroquia 5. obtener información de las parroquias 6. ingresar producto agrícola 7. modificar producto agrícola 8. eliminar producto agrícola 9. seleccionar producto agrícola 10. obtener productos agrícolas 11. ingresar producto pecuario 12. modificar producto pecuario 13. eliminar producto pecuario 14. seleccionar producto pecuario 15. obtener productos pecuarios 16. ingresar infraestructura gubernamental 17. modificar infraestructura gubernamental 18. eliminar infraestructura gubernamental 19. seleccionar infraestructura gubernamental 20. obtener infraestructura gubernamental 21. ingresar estación de comercio 22. modificar estación de comercio 23. eliminar estación de comercio 24. seleccionar estación de comercio 25. obtener estaciones de comercio 26. ingresar sitio turístico 27. modificar sitio turístico 28. eliminar sitio turístico

Realizado por: Alex Cunachi y Erick Dávalos, 2022

3.6. Planteamiento de la hipótesis

H0: El uso de buenas prácticas de seguridad en el mantenimiento NO influye en la vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

H1: El uso de buenas prácticas de seguridad en el mantenimiento influye en la vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

3.6.1. Operacionalización de variables

Se identificó variables independiente y dependiente definidas a continuación:

Independiente: Buenas prácticas de seguridad en el mantenimiento.

Dependiente: Vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

Tabla 4-3: Operacionalización de variables

Hipótesis	Variable	Indicadores	Instrumentos
El uso de buenas prácticas de seguridad en el mantenimiento, influye en la vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo	Buenas prácticas de seguridad en el mantenimiento	Técnicas de Aseguramiento aplicadas	Entorno de desarrollo integrado Código fuente
	Vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo	Confidencialidad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos
		Integridad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos
		Disponibilidad	Software de monitoreo (NMAP, Burpsuite) Ejecución de algoritmos maliciosos

Realizado por: Alex Cunachi y Erick Dávalos, 2022

3.7. Aplicación de la metodología Mantema en el sistema de información productiva CONAGOPARE Chimborazo

En el presente trabajo de titulación se está realizando mantenimiento de software se necesita de una metodología que se encargue de un conjunto de técnicas, métodos y tareas que permita diseñar una solución para la misma, por lo cual se determinó la metodología Mantema, como se definió anteriormente en la macroestructura de la **Ilustración 4-2** en el capítulo de fundamentos teóricos en donde se determinó las actividades y tareas iniciales, actividades y tareas del planificable debido a que no se tendrá que realizar algún mantenimiento de tipo no planificable y se finaliza con actividades y tareas finales, la metodología permite realizar el proceso de manera cíclica

dependiendo de la necesidad de cambios. A continuación, se presenta el procedimiento de las actividades y tareas de la metodología Mantema.

3.7.1. Fase 1: Identificación, clasificación y priorización del problema

En esta fase se deben identificar, clasificar y priorizar las solicitudes de modificación del software, siguiendo las siguientes actividades y tareas:

3.7.1.1. Actividades y tareas iniciales

a) Actividad I0. Estudio inicial

- **Tarea I0.1 Iniciar y recoger información**

Como primer requisito el solicitante realizó la entrega de la solicitud de modificación la cual se encuentra en el (**Anexo A - Sección 1**), misma que es recibida y registrada por el gestor de peticiones en un repositorio para su control.

De lo cual se determinó los cambios de implementación de seguridad mediante la aplicación de las buenas prácticas, mantenimiento de tipo correctivo y perfectivo debido a que se realizó un aumento en las funcionalidades y corrección de errores.

b) Actividad I1. Planificación del proceso

- **Tarea I1.1 Planificar calendario y responsables**

Se generó un calendario de reuniones y se enumeró los interlocutores válidos y los responsables de ambas partes. Las reuniones planificadas sirvieron como hitos de comprobación de resultados. Esto permitió tener un estimado del tiempo que realizó el proyecto de mantenimiento con las modificaciones, corrección de errores, aumento de funcionalidad, actualización de documentos, implementación de seguridad mediante buenas prácticas y pruebas del sistema el documento de planificación se encuentra detallado (**Anexo A - Sección 2**).

Debido a que se determinó una planificación, también existió un personal involucrado los cuales estuvieron inmersos en la realización del proyecto de mantenimiento, los involucrados y sus roles se los puede ver en **Tabla 5-3**.

Tabla 5-3: Personal involucrado

Persona	Contacto	Rol
Ing. Marco Álvarez	Conagoparechimborazo.gob.ec	Técnico del sistema
Ing. Gloria de Lourdes Arcos Medina	Gloria.arcos@esPOCH.edu.ec	Revisores del proyecto
Ing. Oscar Danilo Gavilánez Álvarez	Oscar.gavilanez@esPOCH.edu.ec	
Alex Santiago Cunachi Chacha	alex.cunachi@esPOCH.edu.ec	Analista, diseñador, programador, tester
Erick Fabricio Dávalos Mancero	erick.davalos@esPOCH.edu.ec	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Tarea I1.2 Adquirir conocimiento de la aplicación**

En esta tarea, el equipo de mantenimiento estudió la documentación existente del software objeto de la prestación del servicio, el código de los programas, se realizó entrevistas con el encargado del sistema con el fin de obtener la información que se necesitó para continuar con la realización del proyecto, etc. Durante esta tarea se realizó una recolección de los documentos actuales, para lo cual se realizó el análisis de los mismo y así comprender, la arquitectura establecida, el lenguaje de programación con la cual está desarrollada y las herramientas utilizadas, por lo que se obtuvo un inventario de todos los documentos pertinentes al sistema antiguamente desarrollado, el cual se encuentra detallado en (**Anexo A - Sección 3**)

c) Actividad I2. Análisis de la petición de modificación

- **Tarea I2.1 Recibir petición de modificación**

El usuario, entregó una petición de modificación que fue recibida y registrada por el gestor de peticiones, que le asigna un identificador único.

Se realizó la petición al grupo de desarrollo y se emitió la solicitud de manera autorizada una vez realizado el análisis, firmado y aprobado por los desarrolladores en este caso el equipo de trabajo encargado del mantenimiento de software al sistema de información productiva CONAGOPARE Chimborazo.

- **Tarea I2.2 Decidir tipo de mantenimiento**

A partir de la petición de modificación recibida y registrada en la **Tarea I2.1**, el gestor de peticiones decidió aceptar la petición de modificación, para este caso se realizó el proceso de decisión de tipo de mantenimiento que debió aplicarse en este proyecto. La petición de modificación fue entregada al planificador que la distribuye y encauza al tipo de mantenimiento que fue correspondido. Por consiguiente, la petición de modificación pasó a la siguiente fase lo

cual se determinó que el tipo de mantenimiento de software a realizó en el proyecto fue de tipo perfectivo y correctivo dicho proceso se encuentra detallado en **(Anexo A - Sección 4)**.

3.7.2. Fase 2: Análisis

En esta fase se debe diagnosticar y analizar el cambio para dar respuesta a la solicitud de mantenimiento de software que ha sido aprobada en la fase anterior en la cual se obtuvo información necesaria para saber si el proyecto es factible, siguiendo las siguientes actividades:

3.7.2.1. Actividades y tareas del mantenimiento planificable

Para continuar con el proceso de mantenimiento se debió revisar los resultados obtenidos de los documentos de la fase 1 la solicitud de modificación, y el tipo de mantenimiento que se aplicó.

Los requisitos para aplicar estas actividades y tareas son:

- Solicitud de modificación clasificada por el tipo de mantenimiento correctivo, perfectivo, preventivo o adaptativo.
- Solicitud de modificación aprobada.

a) Actividad P1. Análisis de la petición

- **Tarea P1.1 Valorar petición**

La Petición de modificación que fue recibida en la Tarea I2.1 del conjunto de actividades y tareas iniciales comunes y distribuida en la Tarea I2.2 está ya en posesión del equipo de mantenimiento que va a llevó a cabo la intervención.

En esta tarea, se valoraron los costes y esfuerzos necesarios para servir la petición. Igualmente, se valoraron la disponibilidad de recursos y de calendario para estimar los plazos de la intervención. Para lo cual se realizó un proceso de gestión de riesgos para analizar posibles circunstancias que podrían causar un retraso en el proceso del mantenimiento de software del sistema de información productiva CONAGOPARE Chimborazo por lo que a continuación se detalla lo pertinente al análisis

- **Análisis y gestión de riesgos**

Se realizó una identificación de posibles riesgos existentes por lo cual se obtuvo que tiene un total de 9 posibles riesgos distribuidos en cinco riesgos de proyecto, dos riesgos técnicos y dos riesgos de negocio. Luego se realizó un análisis de los riesgos para obtener su prioridad, y por último se

realizó la gestión para disminuir probabilidad de ocurrencia y aumentar el cumplimiento de los objetivos propuestos en el proyecto. Todo el proceso de la gestión de riesgos se lo puede observar detalladamente en la (**Anexo A - Sección 5**).

- **Estudio de factibilidad**

Para determinar costos, recursos y personal necesario se realizó mediante un análisis de factibilidad esto con la finalidad de determinar si el mantenimiento de software al sistema es factible con base a los aspectos necesarios para la ejecución del proyecto los cuales consta de:

- **Factibilidad técnica**

Debido a que se implementó mantenimiento de software de tipo perfectivo, se va a desarrollar algunas funcionalidades por lo que se realizó un estudio de factibilidad técnica en lo cual se definió el hardware y software necesarios para el desarrollo, la instalación e implementación del sistema web. Las herramientas lenguajes utilizados en este trabajo de integración curricular son de licencia libre lo cual resulta beneficioso en la disminución de costos.

- **Factibilidad operativa**

Se determinó qué tan factible es la operación o manipulación del sistema por parte de los usuarios, este proceso se lo realizó mediante un estudio de factibilidad operativa aquí se detalló el personal y la maquinaria necesaria para que el sistema siga manteniendo su funcionalidad después de ponerlo en marcha luego de haber aplicado el proceso de mantenimiento de software.

- **Factibilidad económica**

Para determinar todos los costos necesarios que se requiere en el proceso de mantenimiento de software del sistema de información productiva CONAGOPARE Chimborazo, se realizó mediante el análisis de la factibilidad económica para esto se determinó: costo de personal, instalación, operación, materiales y suministros, gastos adicionales que son necesarios para obtener un recurso económico real y así poder tener una versión más clara de si el proyecto llega a ser factible por la parte económica.

Todo el proceso del estudio de factibilidad técnica, operativa y económica se encuentra debidamente detallada en el (**Anexo A - Sección 6**).

- **Tarea P1.2 Documentar posibles soluciones y elegir alternativa adecuada**

Se documenta la causa del error y se indican las posibles alternativas de implementación en el documento de diagnóstico del error y posibles soluciones dejando pendiente de rellenar la alternativa de implementación elegida, se encuentra detallado en (**Anexo A - Sección 7**).

3.7.3. Fase 3: Diseño y desarrollo

En esta fase se debe determinar qué parte del sistema software se ve afectado, dejando claramente definido y documentado qué componentes hay que modificar siguiendo las siguientes actividades.

3.7.3.1. Actividad P2. Intervención y pruebas

El presente trabajo de integración curricular al tener como objetivo principal el mantenimiento de software a un sistema web, los estándares de codificación, el diseño de las interfaces y la base de datos ya se encuentran establecidos, por tanto, para continuar con el desarrollo mediante mantenimiento perfectivo se lo va a realizar con lo que ya ha estado propuesto en la documentación previamente analizada.

- **Tarea P2.3 Ejecutar intervención**

El equipo de mantenimiento ejecutó las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada para ello se analizó las herramientas propuestas o la documentación pertinente al sistema desarrollado y se obtiene lo siguiente:

3.7.3.2. Análisis y rediseño de la base de datos

El sistema se encuentra desarrollado con el gestor de base de datos de MySQL, por lo que decidió mantener el mismo gestor para el rediseño de la base de datos debido a que hubo una adicción de las funcionalidades que fueron propuestas por el administrador de la aplicación. Para el rediseño se realizó un análisis a la base de datos actual obteniendo así la información de que está compuesta de 14 tablas relacionadas entre sí, se realizó este análisis con la finalidad de comprender la estructura de la base de datos y así poder desarrollar los cambios o el rediseño de la manera más óptima y eficaz. El modelo entidad-relación y modelo físico de la base de datos del sistema desarrollado anteriormente se encuentra detallado en el (**Anexo A - Sección 8**).

3.7.3.3. Análisis de la arquitectura del sistema

Uno de los objetivos propuestos en el presente trabajo de integración curricular es analizar la arquitectura con la que está desarrollada el sistema por lo que se obtiene lo siguiente:

a) Descripción general del software

El software se llama "sistema de información productiva agropecuaria CONAGOPARE Chimborazo" y está desarrollado en el patrón de arquitectura MVC (Model-View-Controller). Está diseñado para funcionar en plataformas web modernas. Su propósito principal es la de proporcionar información detallada de las actividades de las parroquias que conforman la provincia de Chimborazo; además de ofrecer una interfaz de usuario intuitiva y fácil de usar para llevar a cabo tareas específicas relacionadas con el manejo de datos.

b) Visión general de la arquitectura

La arquitectura del software está diseñada de manera modular, con diferentes módulos encargados de realizar tareas específicas y una capa de controladores que gestiona las interacciones entre ellos. El patrón de diseño utilizado es MVC, que permite separar las responsabilidades de la lógica de negocio, la interfaz de usuario y la gestión de datos. En cuanto a las tecnologías utilizadas, el front-end está desarrollado en angular y utiliza componentes de material design para ofrecer una interfaz de usuario atractiva y consistente. El back-end está desarrollado en PHP para facilitar la implementación de la lógica de negocio y la gestión de datos. La base de datos está desarrollada en el gestor de base de datos MySQL mismo que permite la creación de tablas relacionales, con el fin de guardar y obtener la información que es requerida por la institución pública de CONAGOPARE Chimborazo.

c) Descripción detallada de la arquitectura

La arquitectura del software consta de tres capas principales: la capa de vista, la capa de controlador y la capa de modelo. La capa de vista está compuesta por componentes de Angular que se encargan de mostrar la interfaz de usuario y recibir la entrada del usuario. La capa de controlador está compuesta por controladores de PHP que gestionan las peticiones HTTP y se encargan de coordinar la ejecución de la lógica de negocio y la interacción con la base de datos. La capa de modelo es la que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

En general, la arquitectura MVC permite separar los datos y la lógica de negocio del sistema de la interfaz de usuario, lo que facilita el mantenimiento y el desarrollo del sistema. En el caso del sistema descrito, Angular se utiliza para implementar la vista y el controlador, permitiendo a los desarrolladores construir interfaces de usuario ricas y responsive que se ejecutan en el navegador

del usuario. PHP se utiliza para implementar el modelo, llevando a cabo las operaciones de acceso a datos y lógica de negocio del sistema. MySQL se utiliza como base de datos para almacenar y gestionar la información del sistema. La arquitectura descrita del sistema desarrollado se lo muestra en la **Ilustración 1-3**.

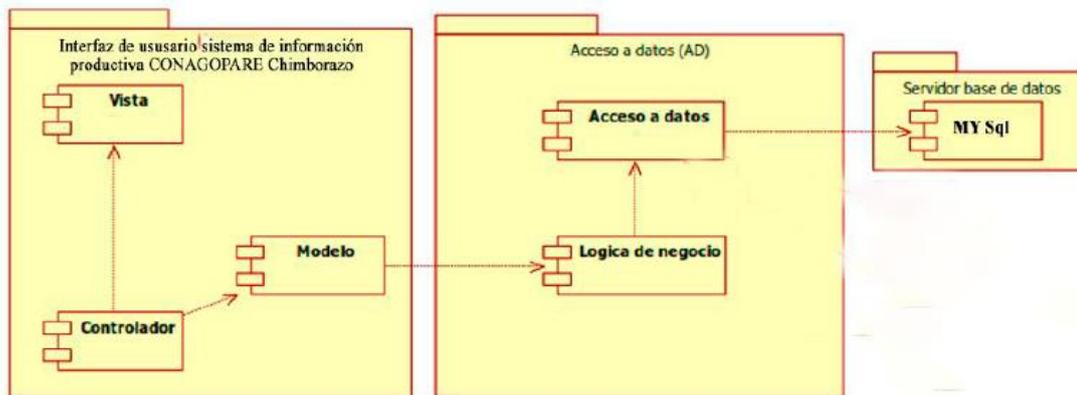


Ilustración 1-3: Diagrama de componentes modelo-vista-controlador del sistema
Realizado por: Alex Cunachi y Erick Dávalos, 2022

En la arquitectura del software, los componentes de Angular se encargan de mostrar la interfaz de usuario y recibir la entrada del usuario. Cuando el usuario realiza una acción que requiere la ejecución de lógica de negocio, el componente envía una petición HTTP al Router de Angular, que se encarga de enrutar al controlador adecuado de PHP. El controlador recibe la petición, ejecuta la lógica de negocio necesaria y, si es necesario, interactúa con la base de datos. Una vez finalizada la ejecución, el controlador envía una respuesta HTTP al componente de Angular, que se encarga de actualizar la interfaz de usuario en consecuencia.

d) Análisis de las ventajas y desventajas de la arquitectura

• Ventajas

- Una de las principales ventajas de la arquitectura del software es que permite separar claramente las responsabilidades de la lógica de negocio, la interfaz de usuario y la gestión de datos. Esto facilita la mantenibilidad y el testeado del software, ya que cada una de las capas puede ser desarrollada y probada de manera independiente.
- Además, el patrón MVC permite utilizar diferentes tecnologías en cada capa, lo que facilita la integración de nuevas tecnologías y la reutilización de componentes existentes.
- Permite trabajar de forma independiente en diferentes capas del sistema, lo que facilita el trabajo en equipo y acelera el desarrollo.
- Además, el uso de Angular en el front-end permite crear interfaces de usuario atractivas y de fácil uso, lo que puede mejorar la experiencia del usuario final.

- **Desventajas**

- La utilización de diferentes tecnologías (Angular, PHP, Mysql) puede requerir que el equipo de desarrollo tenga un conocimiento especializado en cada uno de ellos, lo que puede aumentar los costos de capacitación y el tiempo de desarrollo
- El uso de diferentes tecnologías puede requerir una mayor cantidad de tiempo y esfuerzo para configurar y mantener el entorno de desarrollo y producción.

El sistema de información productiva agropecuaria CONAGOPARE Chimborazo diseñado para proporcionar información detallada sobre las actividades de las parroquias de la provincia de Chimborazo utiliza el patrón de arquitectura MVC. El patrón MVC permite separar las responsabilidades de la lógica de negocio, la interfaz de usuario y la gestión de datos, lo que facilita el mantenimiento y el desarrollo del sistema. Además, la arquitectura es modular, lo que permite una mayor escalabilidad y flexibilidad en el sistema; aunque puede requerir un conocimiento especializado en las tecnologías utilizadas, y puede ocasionar una leve dificultad en la depuración y la solución de problemas, estas desventajas son fácilmente superadas por las ventajas que proporciona la utilización de esta arquitectura.

En resumen, el patrón MVC es una excelente opción para sistemas complejos y con un alto nivel de interacción con el usuario, donde sea necesaria una separación clara de responsabilidades y una buena gestión de datos. A pesar de las posibles desventajas, se recomienda utilizar esta arquitectura ya que ofrece una gran ventaja en cuanto a la organización y mantenimiento del sistema.

- **Puntos estimados por método T-Shirt**

Se obtuvo estimaciones con el fin de organizar las tareas y darle más relevancia a aquellas funcionalidades que son de mayor prioridad, para ello se realizó el método de la talla de la camiseta o T-shirt, de las cuales las tallas o estimaciones del método son S, M, L, XL como se presenta en la **Tabla 6-3**.

Tabla 6-3: Estimaciones por el método T-Shirt

Talla	Puntos estimados	Horas de Trabajo
S	30	30
M	60	60
L	120	120
XL	240	240

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Este cuadro de estimaciones es de utilidad para priorizar puntos estimados y horas de trabajo para realizar las funcionalidades que se aumentan en el sistema así aplicar mantenimiento perfectivo y son detalladas en historias de usuario.

- **Implementación**

Debido a que se va a realizar un mantenimiento de software de tipo perfectivo del sistema y eso conlleva al aumento de las funcionalidades por este motivo el equipo de trabajo detalló lo que se ha ido realizando los cambios que se va obteniendo, mediante historias de usuario por lo que el modelo se detalla en la **Tabla 7-3** y el resto de las historias de usuario desarrolladas se encuentran descritas en la (**Anexo A - Sección 9**)

Tabla 7-3: Modelo de historia de usuario

Historia de Usuario	
Número: HU_01	Nombre de la Historia: Gestión del módulo de empresa productiva
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período Asignado: 2
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de una empresa productiva, para tener conocimiento de su nombre y a que se dedica	
Observaciones:	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mantema es una metodología utilizada para el proceso de mantenimiento de software y que se aplicó en el presente trabajo de integración curricular, permite una comunicación con el cliente de manera recurrente, con la finalidad de cumplir con los objetivos propuestos en base al cronograma, esto se lo realiza para tener una aceptación por parte del cliente de lo desarrollado y así cumplir con sus expectativas.

- **Pruebas del sistema**

Después de determinar las historias de usuario del desarrollo de las funcionalidades se necesita su respectiva prueba de aceptación con la finalidad de obtener de manera exitosa el cambio en el sistema para lo cual se requiere de un modelo que se lo encuentra en la **Tabla 8-3** y el restante de las pruebas de aceptación se lo encuentra en la (**Anexo A - Sección 9**).

Tabla 8-3: Modelo de prueba de aceptación

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_01	Nombre de la Historia: Gestión del módulo de empresa productiva
Nombre: Verificar que se ingrese, modifique, elimine y visualice la empresa productiva por parroquias de Chimborazo	
Responsable: Alex Cunachi	Fecha: 09/05/2022
Descripción: Verificar que el módulo de empresas productivas se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de empresa productiva ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de la empresa productiva de la parroquia y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la empresa a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará 	
Evaluación de la Prueba: Exitosa	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Tarea P2.4 Ejecutar pruebas unitarias**

El equipo de mantenimiento realizó las pruebas unitarias sobre el producto software intervenido. Se comprobó que la petición de modificación queda servida. una vez finalizadas, se generó el documento de pruebas unitarias realizadas.

En la (**Anexo A - Sección 10**) se detallan los resultados de las pruebas unitarias por funcionalidad realizadas en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

- **Tarea P2.5 Ejecutar pruebas de integración**

El Equipo de mantenimiento comprobó que los diferentes elementos del software funcionan correctamente de forma conjunta.

Para demostrar que los sistemas están funcionales y cumplen con lo pactado al inicio del trabajo de integración curricular se detalla los resultados de las pruebas de integración que se puede observar en la detalladamente (**Anexo A - Sección 10**).

- **Tarea P2.7 Verificar y validar corrección con el cliente**

El equipo de mantenimiento y la Organización del sistema (o el perfil en que se haya delegado) se reunieron para comprobar que el producto intervenido funciona correctamente. Por lo cual se

realizó una reunión con los integrantes de la parroquia obteniendo resultados favorables en cuanto a las mejoras en el sistema en comparación con el sistema antiguo o antes de haber realizado el mantenimiento de software. Los resultados de la comparativa de la interfaz del sistema están detallados en la (**Anexo A - Sección 11**).

- **Tarea P2.8 Re documentar manual de usuario**

El equipo de mantenimiento re documentó el manual de usuario, aportando una nueva perspectiva en cuanto a su funcionamiento y la utilización del sistema. En dicho manual se agregó la manera de utilizar las nuevas funcionalidades que se incrementaron en el mantenimiento perfecto, además de la gestión documental oficinas que se envían a la institución pública y los memorándums que se envían entre los integrantes de CONAGOPARE Chimborazo con el fin de responder favorablemente a los oficios emitidos, y el manejo de los usuarios.

3.7.4. Fase 4: Finalización del proceso de mantenimiento de software

Después de haber aplicado el proceso de mantenimiento de software mediante el uso de la metodología propuesta, se aplicó pruebas sobre el sistema lanzado en el servidor, para lo cual se evaluaron todas las 57 funcionalidades del sistema de información productiva agropecuaria CONAGOPARE Chimborazo con la finalidad de realizar la verificación del funcionamiento de dicho sistema; por consiguiente para el lanzamiento del sistema web se notificó al Ing. Marco Álvarez encargado de la aplicación el link para el acceso y se realizó una copia de seguridad guardado en los ordenadores de desarrollo. Por último, para terminar con el proceso de mantenimiento de software, se realizó, la capacitación al personal que utilizará el sistema de información productiva agropecuaria CONAGOPARE Chimborazo una vez este se encuentre en explotación debido a que los encargados de la aplicación deben tener conocimientos de como manipular el sistema web.

3.7.4.1. Actividades y Tareas Finales Comunes

a) Actividad F2. Actualización de la base de datos histórica

- **Tarea F2.2 Actualizar base de datos**

Los datos recogidos en la tarea anterior son incorporados a la base de datos histórica. Al realizar una actualización de la base de datos se obtuvo un incremento de funcionalidades por ende un incremento en las clases, métodos y atributos que integran la base de datos del sistema de información productiva agropecuaria CONAGOPARE Chimborazo. Se actualizó la base de datos incrementando 10 tablas después del mantenimiento de software dando un total de 24 tablas y

realizando la modificación en dos tablas que son: usuario y parroquia, dicha base de datos modificada se encuentra detallada en la (**Anexo A - Sección 12**) mediante el modelo entidad-relación y modelo físico.

Además, también se obtuvo el diccionario de datos con la finalidad de listar los datos organizados en donde se determina de manera rigurosa, las características lógicas y puntuales tales como nombre, tipo de contenido y descripción de negocio, de tal forma que se cuente con un elemento común, para comprender de mejor manera de la base de datos. Dicho diccionario de datos se encuentra debidamente detallado en el (**Anexo A - Sección 13**)

b) Actividad F3. Retirada

- **Tarea F3.1 Desarrollar plan de retirada**

El equipo de mantenimiento redactó un documento en el que describe cuándo se llevará a cabo la retirada del software.

- **Tarea F3.5 Almacenar datos del entorno antiguo**

Se realizó copias de seguridad del sistema y documentación antigua o antes de realizar el mantenimiento de software con el fin de determinar mejoras en el sistema y comparativas para obtener resultados favorables, y también actualizar los documentos como el manual de usuario y manual técnico pertinentes al sistema de información productiva CONAGOPARE Chimborazo.

c) Actividad F4. Fin de la externalización

- **Tarea F4.1 Entrega del inventario y de la documentación**

Dependiendo de lo que se pactara en el contrato, posiblemente la organización de mantenimiento se encuentre obligada a entregar al cliente todos los productos software generados y modificados durante el periodo en que ha sido responsable del mantenimiento.

- **Tarea F4.3 Cesión definitiva del servicio**

La organización de mantenimiento deja de prestar sus servicios al cliente con carácter definitivo.

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos al aplicar las metodologías y herramientas de software propuestas, con el fin de lograr los objetivos que se planteó al inicio del presente trabajo de integración curricular.

4.1. Resultados del indicador de funcionalidad

Mediante la ISO/IEC 25010 se pudo investigar los indicadores que pertenecen a la Funcionalidad de un sistema software por lo cual en este capítulo se demuestran los resultados obtenidos por completitud funcional, exactitud funcional y pertinencia funcional mediante las fórmulas a aplicar.

4.1.1. Resultados obtenidos de los errores de funcionalidad en el sistema

- **Errores obtenidos antes del mantenimiento de software**

Como se puede notar en la **Tabla 1-4** con una X se detalla los errores que se encontraron en el sistema antes del mantenimiento de software y luego de haber sido manipulado por los usuarios o técnicos de las parroquias de Chimborazo por un tiempo de 30 minutos obteniendo los siguientes resultados:

- Todos los técnicos encontraron en el módulo producción agrícola y en el módulo de producción pecuaria errores en la función de ingresar y modificar.
- Los técnicos 2, 6 y 10 encontraron errores en la parte de interfaz de usuario.
- Los técnicos 1, 4, 7, 9 y 11 encontraron errores en la parte de Inicio de sesión o Login.
- En conclusión, antes de aplicar el mantenimiento de software se encontraron 7 errores totales en las funcionalidades que componen los módulos del sistema de información productiva CONAGOPARE Chimborazo.

Tabla 1-4: Resultados de los usuarios en la parte de funcionalidad del sistema, mediante la manipulación por 30 minutos antes del mantenimiento de software

Módulo	Técnico 1	Técnico 2	Técnico 3	Técnico 4	Técnico 5	Técnico 6	Técnico 7	Técnico 8	Técnico 9	Técnico 10	Técnico 11
Módulo de registro o autenticación de datos											
Módulo de información de la parroquia											
Módulo de producción agrícola	X	X	X	X	X	X	X	X	X	X	X
Módulo de producción pecuaria	X	X	X	X	X	X	X	X	X	X	X
Módulo de información gubernamental											
Módulo de estación de comercio											
Módulo de turismo											
Interfaz de usuario		X				X				X	
Login o inicio de sesión	X			X			X		X		X

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Errores obtenidos después del mantenimiento de software**

Después de llevar a cabo el proceso de mantenimiento de software al sistema de información productiva agropecuaria CONAGOPARE Chimborazo, los técnicos de las parroquias realizaron la prueba de manipulación por un total de 30 minutos al sistema, se observó una reducción significativa en los errores encontrados antes del mantenimiento de software. En conclusión, se logró reducir los 7 errores encontrados previamente a un total de 0 errores lo que tiene una equivalencia del 0%

Los datos obtenidos sobre la cantidad de errores en la funcionalidad del sistema son de utilidad para realizar los cálculos de las fórmulas para completitud, exactitud y pertinencia funcional que presentan a continuación.

4.1.2. Resultados obtenidos de completitud funcional

Se determina los valores mediante la fórmula

$$X=A/B$$

A=Número de funciones que están incorrectas o que no fueron implementadas

B= Número de funciones establecidas en la especificación de requisitos

Antes del realizar el mantenimiento

Resultados:

$$X= 7 / 32 = 0,22$$

Equivalencia de un 22% de errores en cuánto al indicador de completitud funcional en sistema web

Después de realizar el mantenimiento

Resultados:

$$X = 0 / 57 = 0$$

Equivalencia de un 0% de errores en cuánto al indicador de completitud funcional en sistema web.

En la **Ilustración 1-4** se muestra la comparativa en el indicador de completitud funcional antes y después de haber realizado el mantenimiento de software teniendo los siguientes resultados:

- Se obtuvo un resultado de 7 de errores que equivalen a un 22%, con un total de 32 funcionalidades realizadas antes del mantenimiento de software.
- Después de haber realizado el mantenimiento de software con un total de 57 funcionalidades realizadas se obtuvo un resultado de 0 errores que equivale al 0%.

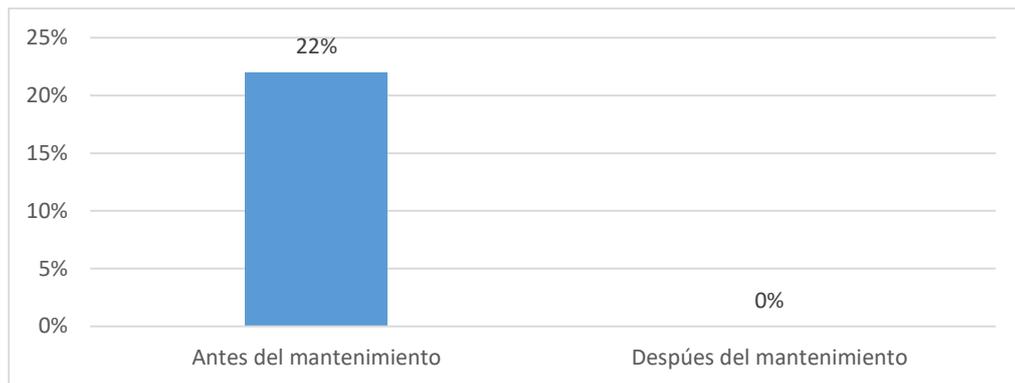


Ilustración 1-4: Comparativa de los resultados de completitud funcional
Realizado por: Alex Cunachi y Erick Dávalos, 2022

4.1.3. Resultados obtenidos de exactitud funcional

Se calcula mediante la fórmula

$$X=A/T$$

A=Número de cálculos inexactos encontrados

T=Tiempo de operación

Antes del mantenimiento

$$X= 0/30 \text{ min} =0$$

Después del mantenimiento

$$X= 0/30 \text{ min} =0$$

En este indicador exactitud funcional se obtuvo un 100% de aprobación de sus resultados antes y después de haber realizado el mantenimiento debido a que al ser utilizado el software por un tiempo de 30 minutos no se encontraron datos inexactos por lo que su resultado es 0 errores.

4.1.4. Resultados obtenidos de pertinencia funcional

Se determina los valores mediante la fórmula:

$$X=A/B$$

A=Número de funciones que se detectan problemas en evaluación

B=Número de funciones evaluadas

Antes del mantenimiento

$$X= 7/32= 0,22$$

Equivalencia de un 22% de errores en cuánto al indicador de pertinencia funcional en sistema web.

Después del mantenimiento

$$X = 0/57 = 0$$

Equivalencia de un 0% de errores en cuanto al indicador de pertinencia funcional en sistema web.

En la **Ilustración 2-4** se muestra la comparativa en el indicador de pertinencia funcional antes y después de haber realizado el mantenimiento de software teniendo los siguientes resultados:

- Se obtuvo un resultado de 7 de errores que equivalen a un 22%, con un total de 32 funcionalidades realizadas antes del mantenimiento de software.
- Después de haber realizado el mantenimiento de software con un total de 57 funcionalidades realizadas se obtuvo un resultado de 0 errores que equivale al 0%.

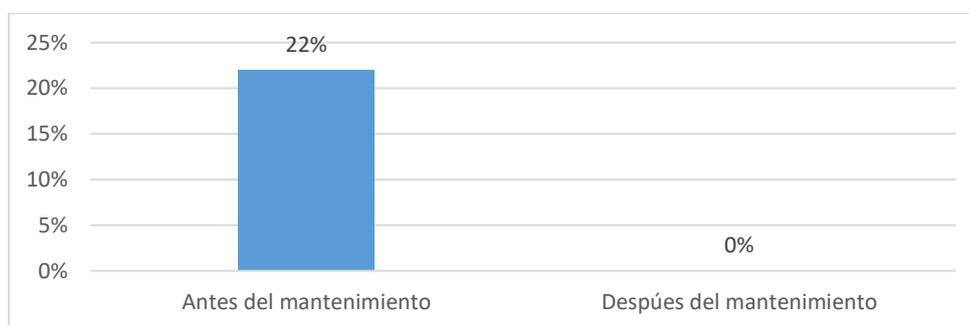


Ilustración 2-4: Comparativa de los resultados de errores de pertinencia funcional
Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mediante la métrica se estableció un grado de ponderación de los indicadores que componen la funcionalidad, se estableció una ponderación de escala de 10 puntos detallados en la **Tabla 2-4**.

Tabla 2-4: Ponderación de prioridad de los indicadores de funcionalidad

Indicador	Grado de importancia	Ponderación
Complejidad Funcional	Media	3
Exactitud Funcional	Media	3
Pertinencia Funcional	Alta	4

Realizado por: Alex Cunachi y Erick Dávalos, 2022

4.1.5. Resultados generales de funcionalidad antes y después de aplicar el proceso de mantenimiento de software

En la **Tabla 3-4** se describe los resultados generales obtenidos antes de haber realizado el mantenimiento de software aplicando las fórmulas pertinentes de cada indicador de funcionalidad, mediante una ponderación sobre 10 puntos se obtuvo lo siguiente:

- El valor de la ponderación para el indicador de complejidad funcional fue de 7,8 puntos.

- El valor de la ponderación para el indicador de exactitud funcional fue de 10 puntos.
- El valor de la ponderación para el indicador de pertinencia funcional fue de 7,8 puntos.
- El valor promedio general de los indicadores de funcionalidad antes del mantenimiento de software fue de 8,53 puntos.

Tabla 3-4: Resultados generales de indicadores de funcionalidad antes del mantenimiento de software

Indicador	Umbral(resultados deseados)	Fórmula		Valor	Ponderación del indicador	Promedio	
		A	B/T				
Compleitud Funcional	0	7	32	0,22	7,8	8,53	
Exactitud Funcional	0	0	30 min.	0	10		
Pertinencia Funcional	0	7	32	0,22	7,8		
Total						25,6	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

En la **Tabla 4-4** se describe los resultados generales obtenidos después de haber realizado el mantenimiento de software aplicando las fórmulas pertinentes de cada indicador de funcionalidad, mediante una ponderación sobre 10 puntos se obtuvo lo siguiente:

- El valor de la ponderación para el indicador de completitud funcional fue de 10 puntos.
- El valor de la ponderación para el indicador de exactitud funcional fue de 10 puntos.
- El valor de la ponderación para el indicador de pertinencia funcional fue de 10 puntos.
- El valor promedio general de los indicadores de funcionalidad antes del mantenimiento de software fue de 10 puntos.

Tabla 4-4: Resultados generales de indicadores de funcionalidad después del mantenimiento de software

Indicador	Umbral (resultados deseados)	Fórmula		Valor	Ponderación del indicador	Promedio	
		A	B/T				
Compleitud Funcional	0	0	57	0	10	10	
Exactitud Funcional	0	0	30 min.	0	10		
Pertinencia Funcional	0	0	57	0	10		
Total						30,00	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

En la **Ilustración 3-4** se tiene una comparativa de los resultados obtenidos de la ponderación sobre un límite de 10 puntos por lo que se obtuvo:

- Una mejora en cuanto a completitud funcional y pertinencia funcional de 2,2 puntos del después de haber realizado el mantenimiento de software.

- Para exactitud funcional se tuvieron 10 puntos antes y después del mantenimiento de software en el sistema de información productiva CONAGOPARE Chimborazo.

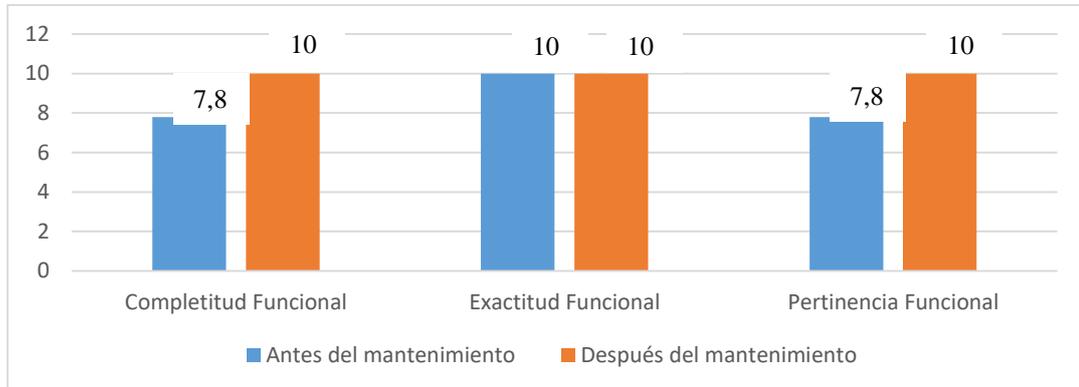


Ilustración 3-4: Comparativa de los resultados de ponderación por indicador
Realizado por: Alex Cunachi y Erick Dávalos, 2022

En el promedio de la ponderación general de funcionalidad de software antes de realizar el mantenimiento se obtuvo un total de 8,53 puntos y después del mantenimiento se obtuvo 10 puntos por lo que se puede demostrar que hubo una mejora de 1,47 puntos en cuanto a funcionalidad después de haber realizado el mantenimiento de software el incremento de su puntaje se presenta en la **Ilustración 4-4**.

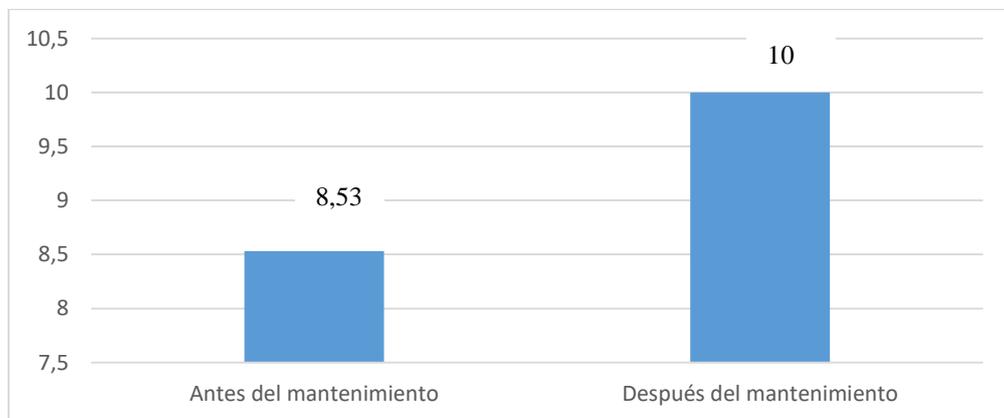


Ilustración 4-4: Comparativa de los resultados de media de funcionalidad
Realizado por: Alex Cunachi y Erick Dávalos, 2022

En conclusión, se obtuvo una mejora en el indicador de funcionalidad demostrando resultados individuales mediante sus sub-indicadores de completitud funcional, exactitud funcional, pertinencia funcional y resultados generales por ponderaciones obtenidas mediante la ISO/IEC 25010.

4.2. Resultados obtenidos del indicador de seguridad

Los resultados obtenidos son de importancia para saber si la aplicación de las buenas prácticas de seguridad en el proceso de mantenimiento de software influye de manera positiva en la minimización de vulnerabilidades en el sistema, para lo cual en el capítulo anterior se definió una metodología y se aplicaron cálculos de muestreo por lo que se determinó realizar un total de 66 ataques de penetración a las funcionalidades desarrolladas antes del mantenimiento de software, y las que se agregaron después del mantenimiento de software con el objetivo de obtener resultados para realizar comparativas estadísticas.

Se determinó realizar los siguientes ataques de penetración al sistema de información productiva agropecuaria CONAGOPARE Chimborazo detallados en la **Tabla 5-4**.

Tabla 5-4: Tipos de ataque de penetración por identificador

ID	OWASP identificador	Tipos de ataques
A1	BROKEN ACCESS CONTROL	A1.1 DIRECTORY TRAVERSAL A1.2 CROSS-SITE REQUEST FORGERY
A2	CRYPTOGRAPHIC FAILURES	A2.1 UNENCRYPTED COMMUNICATION
A3	INJECTION	A3.1 SQL INJECTION A3.2 XSS
A4	IDENTIFICATION AND AUTHENTICATION FAILURES	A4.1 PASSWORD MISMANAGEMENT A4.2 PRIVILEGE ESCALATION A4.3 SESSION FIXATION

Realizado por: Alex Cunachi y Erick Dávalos, 2022

4.2.1. Análisis de resultados de seguridad antes de realizar el proceso de mantenimiento de software

Los valores obtenidos tras ejecutar ataques de penetración al sistema antes de haber realizado el proceso de mantenimiento de software detallado por caso de uso se presentan en la **Tabla 6-4**, los porcentajes obtenidos por ataque a la vulnerabilidad del sistema se detalla en la **Tabla 7-4**. Y los valores detallados por funcionalidad se encuentran desarrollados en el (**Anexo B**).

Tabla 6-4: Resultados por caso de uso del porcentaje de vulnerabilidad en sql Injection antes del mantenimiento de software

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T13	Ingresar información de la parroquia	100% Vulnerable
T14	Modificar información de la parroquia	100% Vulnerable
T15	Ingresar producto agrícola	100% Vulnerable
T16	Modificar producto agrícola	100% Vulnerable

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T17	Ingresar producto pecuario	100% Vulnerable
T18	Modificar producto pecuario	100% Vulnerable
T19	Ingresar infraestructura gubernamental	100% Vulnerable
T20	Obtener infraestructura gubernamental	100% Vulnerable
T21	Ingresar estación de comercio	100% Vulnerable
T22	Modificar estación de comercio	100% Vulnerable
T23	Ingresar sitio turístico	100% Vulnerable
T24	Modificar sitio turístico	100% Vulnerable
Promedio total		100%

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Tabla 7-4: Resultados obtenidos de los ataques antes del mantenimiento de software

Indicador	Ataque	Numero de funcionalidades atacadas	Porcentaje de vulnerabilidad
Confidencialidad	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A2.1 UNENCRYPTED COMMUNICATION	1	100%
	A3.1 SQL INJECTION	12	100%
	A3.2 XSS	12	0%
	A4.1 PASSWORD MISMANAGEMENT	1	100%
	A4.2 PRIVILEGE ESCALATION	30	100%
Integridad	A1.1 DIRECTORY TRAVERSAL	12	0%
	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A3.1 SQL INJECTION	12	100%
	A3.2 XSS	12	0%
	A4.1 PASSWORD MISMANAGEMENT	1	100%
	A4.2 PRIVILEGE ESCALATION	30	100%
Disponibilidad	A1.1 DIRECTORY TRAVERSAL	12	0%
	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A2.1 UNENCRYPTED COMMUNICATION	1	100%
	A3.2 XSS	12	0%

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mediante un diagrama de barras se puede demostrar los resultados obtenidos ante los ataques realizados al sistema antes de aplicar el mantenimiento de software los mismos que reflejan un 100% de vulnerabilidad password mismanagement, unencrypted communication, SQL Injection

y privilege escalation, cross-site request forgery en cuanto a directory traversal se obtiene 0% lo que demuestra que el sistema es muy vulnerable ante ataques de penetración, se lo encuentra en la **Ilustración 5-4**.

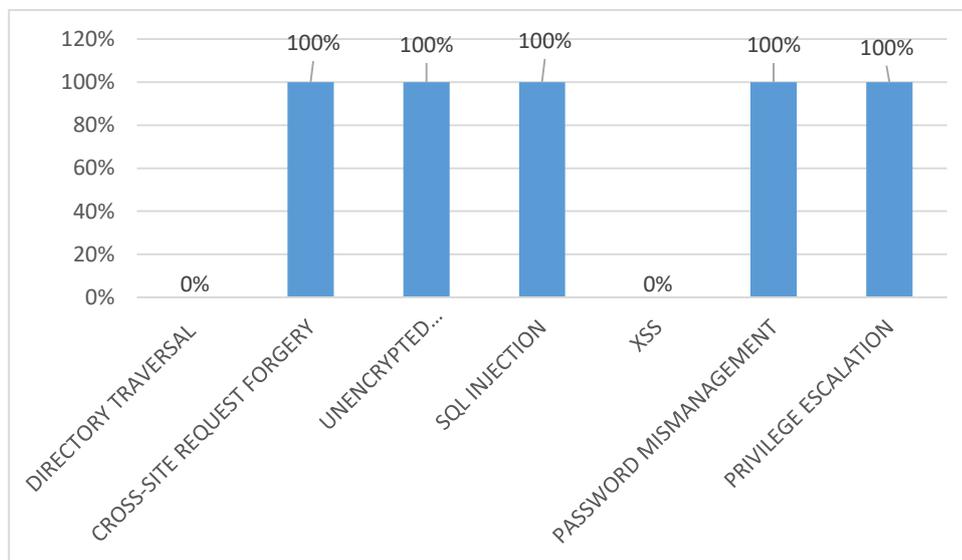


Ilustración 5-4: Porcentaje de error de los ataques realizados al sistema antes del mantenimiento de software

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Los resultados obtenidos antes de aplicar 7 ataques de penetración al sistema de información productiva agropecuaria CONAGOPARE Chimborazo se analizan mediante los parámetros de la estadística descriptiva para obtener valores que puedan ayudar a determinar su conducta.

Tabla 8-4: Resultados de los ataques antes del mantenimiento de software mediante estadística descriptiva

Ataque	Funcionalidades vulnerables sin buenas prácticas (X)
DIRECTORY TRAVERSAL	0
CROSS-SITE REQUEST FORGERY	12
UNENCRYPTED COMMUNICATION	1
SQL INJECTION	12
XSS	0
PASSWORD MISMANAGEMENT	1
PRIVILEGE ESCALATION	30
Media	8,00
Varianza	123,67
Desviación estándar	11,12

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mediante estadística descriptiva se pudo obtener resultados de los ataques realizados al sistema de información productiva agropecuaria CONAGOPARE Chimborazo antes de realizar el

mantenimiento de software obteniendo así información que esta detallada en la Tabla 8-4, y así poder concluir que la desviación estándar tiene un valor de **11,12** variación significativa con respecto a la media **8,00**, los datos tienen muy poca concentración alrededor de la media.

4.2.2. Análisis de resultados de seguridad después de realizar el proceso de mantenimiento de software

Los valores obtenidos tras ejecutar ataques de penetración al sistema después de haber realizado el proceso de mantenimiento de software detallado por caso de uso se presentan en la **Tabla 9-4**, los porcentajes obtenidos por ataque a la vulnerabilidad del sistema se detalla en la **Tabla 10-4**. Y los valores detallados por funcionalidad se encuentran desarrollados en el (**Anexo B**).

Tabla 9-4: Resultados por caso de uso del porcentaje de vulnerabilidad en sql injection después del mantenimiento de software

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T13	Ingresar información de la parroquia	0% Vulnerable
T14	Modificar información de la parroquia	0% Vulnerable
T15	Ingresar producto agrícola	0% Vulnerable
T16	Modificar producto agrícola	0% Vulnerable
T17	Ingresar producto pecuario	0% Vulnerable
T18	Modificar producto pecuario	0% Vulnerable
T19	Ingresar infraestructura gubernamental	0% Vulnerable
T20	Obtener infraestructura gubernamental	0% Vulnerable
T21	Ingresar estación de comercio	0% Vulnerable
T22	Modificar estación de comercio	0% Vulnerable
T23	Ingresar sitio turístico	0% Vulnerable
T24	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Tabla 10-4: Resultados obtenidos de los ataques después del mantenimiento de software

Indicador	Ataque	Numero de funcionalidades atacadas	Porcentaje de vulnerabilidad
Confidencialidad	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A2.1 UNENCRYPTED COMMUNICATION	1	0%
	A3.1 SQL INJECTION	12	0%
	A3.2 XSS	12	0%
	A4.1 PASSWORD MISMANAGEMENT	1	0%
	A4.2 PRIVILEGE ESCALATION	30	0%
Integridad	A1.1 DIRECTORY TRAVERSAL	12	0%

Indicador	Ataque	Numero de funcionalidades atacadas	Porcentaje de vulnerabilidad
	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A3.1 SQL INJECTION	12	0%
	A3.2 XSS	12	0%
	A4.1 PASSWORD MISMANAGEMENT	1	0%
	A4.2 PRIVILEGE ESCALATION	30	0%
Disponibilidad	A1.1 DIRECTORY TRAVERSAL	12	0%
	A1.2 CROSS-SITE REQUEST FORGERY	12	100%
	A2.1 UNENCRYPTED COMMUNICATION	1	0%
	A3.2 XSS	12	0%

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mediante un diagrama de barras se puede demostrar los resultados obtenidos ante los ataques realizados al sistema después de realizar el proceso de mantenimiento de software los mismos que reflejan un 100% de vulnerabilidad en cross-site request forgery en cuanto a password mismanagement, unencrypted communication, SQL Injection, XSS, privilege escalation y directory traversal se obtiene 0% se lo encuentra en la **Ilustración 6-4**.

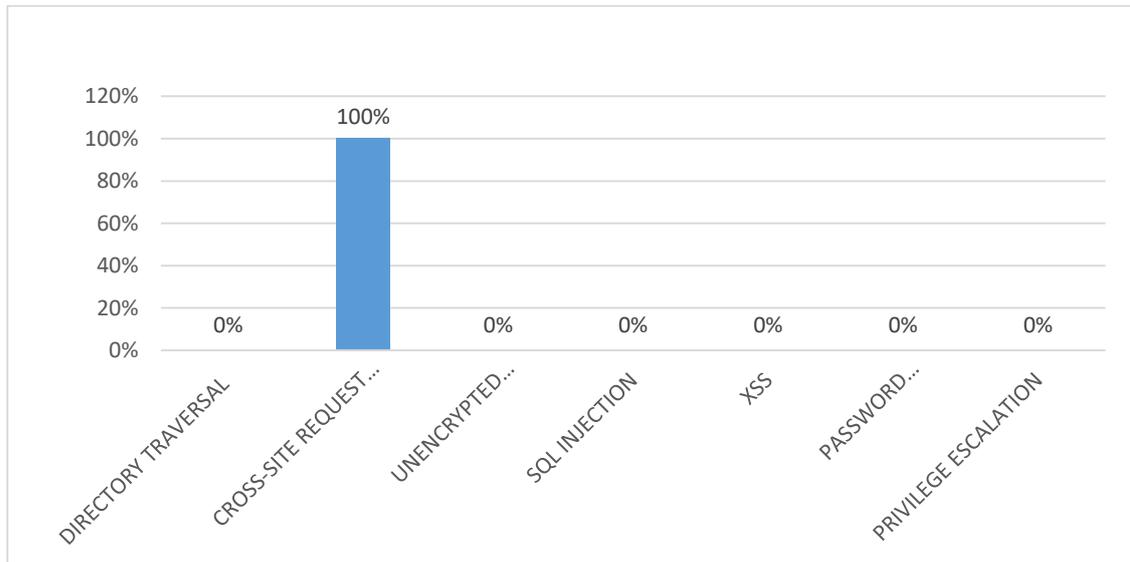


Ilustración 6-4: Porcentaje de error de los ataques realizados al sistema después del mantenimiento de software

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Los resultados obtenidos después de aplicar 7 ataques de penetración al sistema de información productiva agropecuaria CONAGOPARE Chimborazo se analizan mediante los parámetros de la estadística descriptiva para obtener valores que puedan ayudar a determinar su conducta.

Tabla 11-4: Resultados de los ataques después del mantenimiento de software mediante estadística descriptiva

Ataque	Funcionalidades vulnerables sin buenas prácticas (X)
DIRECTORY TRAVERSAL	0
CROSS-SITE REQUEST FORGERY	12
UNENCRYPTED COMMUNICATION	0
SQL INJECTION	0
XSS	0
PASSWORD MISMANAGEMENT	0
PRIVILEGE ESCALATION	0
Media	1,71
Varianza	20,57
Desviación estándar	4,54

Realizado por: Alex Cunachi y Erick Dávalos, 2022

Mediante estadística descriptiva se pudo obtener resultados de los ataques realizados al sistema de información productiva agropecuaria CONAGOPARE Chimborazo después de realizar el mantenimiento de software obteniendo así información que esta detallada en la **Tabla 11-4**, y así poder concluir que la desviación estándar tiene un valor de **4,54** variación significativa con respecto a la media **1,71**, los datos tienen muy poca concentración alrededor de la media.

4.2.3. Comparativo de resultados de seguridad antes y después de aplicar el mantenimiento de software

Una vez implementado las buenas prácticas de seguridad emitidas por la fundación OWASP se concluye realizando el resumen del comportamiento después de haber puesto a prueba el sistema mediante las pruebas de penetración, de lo cual se detalla a continuación una comparativa entre el sistema antiguo y el sistema después de realizar el mantenimiento software **Tabla 12-4**.

Dando que:

P = Pasó las pruebas de ataques con un 0% de vulnerabilidad

F = Falló las pruebas de ataques con un 100% de vulnerabilidad

Tabla 12-4: Comparativa general de los ataques de penetración en el sistema

Ataque	Sistema antes de realizar el mantenimiento	Sistema después de realizar el mantenimiento
A1.2 DIRECTORY TRAVERSAL	P	P
A1.3 CROSS-SITE REQUEST FORGERY	F	F
A2.1 UNENCRYPTED COMMUNICATION	F	P
A3.1 SQL INJECTION	F	P
A3.2 XSS	P	P
A4.1 PASSWORD MISMANAGEMENT	F	P
A4.2 PRIVILEGE ESCALATION	F	P

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Confidencialidad e integridad**

En el **Ilustración 7-4** se puede observar una comparación que se realizó entre la media de los resultados obtenidos en los indicadores de confidencialidad e integridad mediante estadística descriptiva del número de vulnerabilidades del sistema sin aplicar buenas prácticas de seguridad y después de aplicar buenas prácticas de seguridad, donde se puede concluir que el mantenimiento de software influyó de manera positiva en los errores de seguridad del sistema, minimizando la vulnerabilidad en los indicadores de confidencialidad e integridad de 9,25 puntos obtenidos en la media a 2,00 después de haber aplicado buenas prácticas de seguridad en el proceso de mantenimiento de software al sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

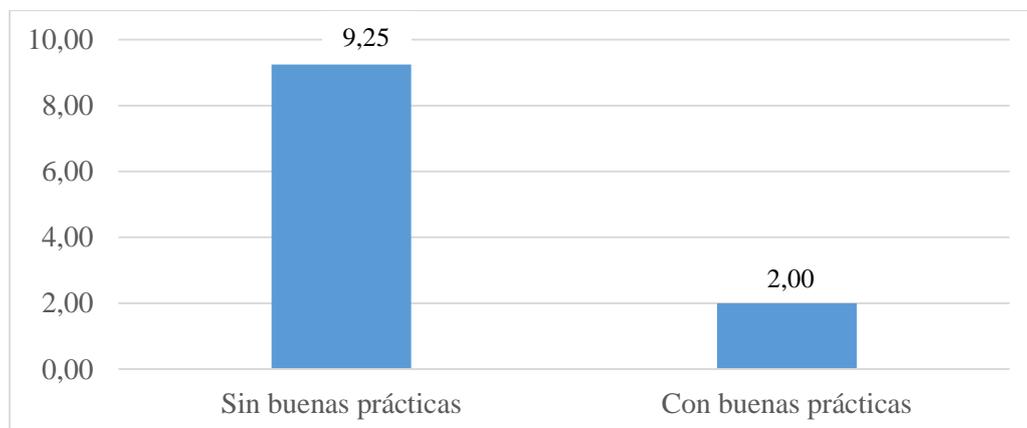


Ilustración 7-4: Confidencialidad e integridad antes y después del mantenimiento

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Disponibilidad**

En la **Ilustración 8-4** se puede observar una comparación que se realizó entre la media de los resultados obtenidos en el indicador de disponibilidad mediante estadística descriptiva del número de vulnerabilidades del sistema sin aplicar buenas prácticas de seguridad y después de aplicar buenas prácticas de seguridad, donde se puede concluir que el mantenimiento de software influyó de manera positiva en los errores de seguridad del sistema, minimizando la vulnerabilidad en el indicador de disponibilidad de 3,25 puntos obtenidos en la media a 3,00 después de haber aplicado buenas prácticas de seguridad en el proceso de mantenimiento de software al sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

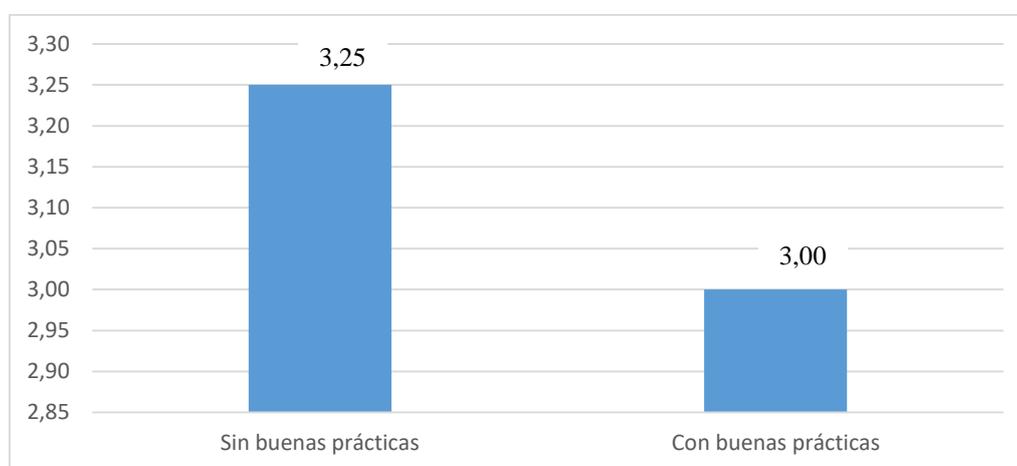


Ilustración 8-4: Disponibilidad antes y después de aplicar buenas prácticas de seguridad en el proceso de mantenimiento de software

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **General**

En el **Ilustración 9-4** se puede observar una comparación que se realizó entre la media de los resultados generales obtenidos mediante estadística descriptiva del número de vulnerabilidades del sistema sin aplicar buenas prácticas de seguridad y después de aplicar buenas prácticas de seguridad, donde se puede concluir que el mantenimiento de software influyó de manera positiva en los errores de seguridad del sistema, minimizando la vulnerabilidad de 8 puntos obtenidos en la media a 1,71 después de haber aplicado buenas prácticas de seguridad en el proceso de mantenimiento de software al sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

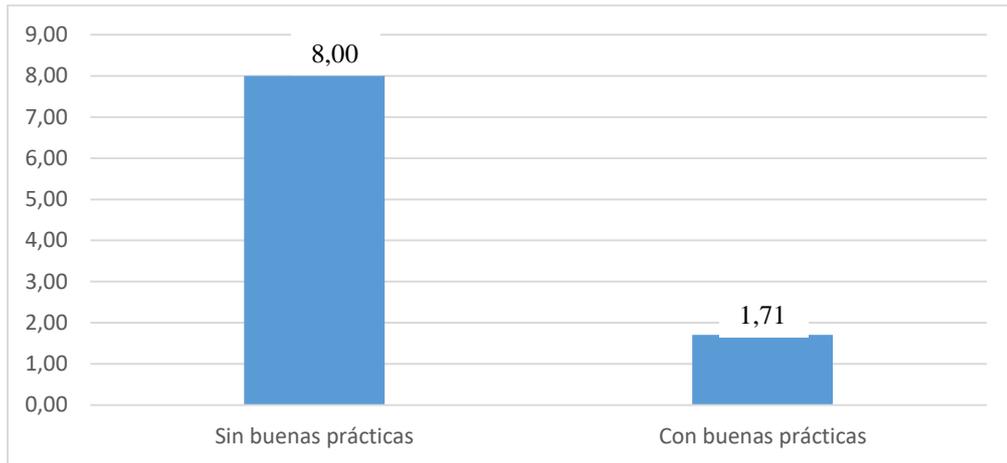


Ilustración 9-4: Resultados generales antes y después de aplicar buenas prácticas de seguridad en el proceso de mantenimiento de software
Realizado por: Alex Cunachi y Erick Dávalos, 2022

En base a la definición de seguridad de la información según ISO 27001, es “la preservación de la confidencialidad, integridad y disponibilidad de la información”. Entonces la seguridad será igual a la suma de la confidencialidad (C), Integridad (I) y la Disponibilidad (D), como lo determina (Paguay, 2015, pp.56-57).

La importancia que radica los indicadores de seguridad en este presente trabajo de integración curricular se lo determinó de la siguiente manera la amenaza a la integridad (I) se representa con 40%, la amenaza a la confidencialidad (C) con 30% por último la amenaza a la disponibilidad (D) un 30%, obteniendo así un total del 100%. Se tiene como prioridad a la integridad debido a que por medio de este indicador la información se debe tener inalterada, y solo se podrá haber cambios en la información si esta tiene permisos.

$$Vulnerabilidad = C + I + D$$

Se realizó una cantidad de ataques al sistema con la finalidad de obtener resultados antes y después de realizar el proceso de mantenimiento de software para determinar valores en porcentajes y así poder realizar comparaciones, para obtener conclusiones de si existe una mejora en cuanto a la minimización de vulnerabilidades en el sistema. Estos resultados se determinan por los indicadores de seguridad de CIA los cuales están detallados en la **Tabla 13-4**.

Por lo que se puede concluir que al aplicar buenas prácticas de seguridad en el proceso de mantenimiento de software se demostró una mejora en la disminución de vulnerabilidades del **47,50%** en el sistema de Información productiva agropecuaria CONAGOPARE Chimborazo

Tabla 13-4: Resumen de resultados por indicadores CIA en el sistema de información productiva agropecuario CONAGOPARE Chimborazo

Categoría		Porcentaje de Vulnerabilidad del sistema sin buenas prácticas		Porcentaje de Vulnerabilidad del sistema con buenas prácticas		Minimización de vulnerabilidad
Indicador	Equivalencia	V. Real	V. Equivalente	V. Real	V. Equivalente	
Confidencialidad	30%	83,33%	25,00%	16,67%	5,00%	66,66%
Integridad	40%	66,67%	26,66%	16,67%	6,66%	50,00%
Disponibilidad	30%	50,00%	15,00%	25,00%	7,50%	25,00%
TOTAL	100%		66,66%		19,16%	47,50%

Realizado por: Alex Cunachi y Erick Dávalos, 2022

En el **Ilustración 10-4**, se muestra los datos obtenidos representados en un diagrama de barras en el cual se tiene que en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo se obtuvo mejora significativa en los indicadores CIA:

- Para confidencialidad existe una mejora del 66,66% pasando de 83,33% de vulnerabilidad antes de aplicar mantenimiento de software a 16,67% de vulnerabilidad después de realizar mantenimiento de software.
- Para integridad existe una mejora del 50,00% pasando de 66,67% de vulnerabilidad antes de aplicar mantenimiento de software a 16,67% de vulnerabilidad después de realizar mantenimiento de software.
- Para disponibilidad existe una mejora del 25,00% pasando de 50,00% de vulnerabilidad antes de aplicar mantenimiento de software a 25,00% de vulnerabilidad después de realizar mantenimiento de software.

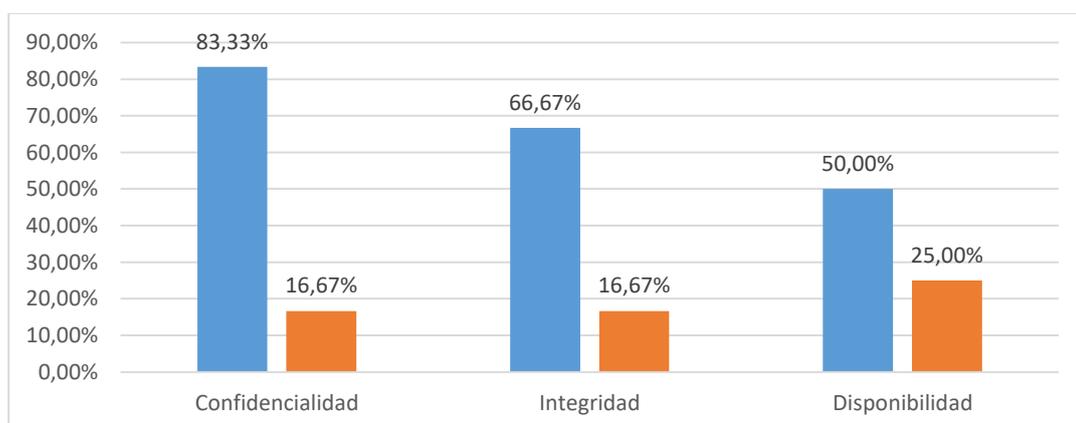


Ilustración 10-4: Comparativa de resultados por indicadores de seguridad (CIA)

Realizado por: Alex Cunachi y Erick Dávalos, 2022

4.2.4. Prueba chi-cuadrado (x2)

Para demostrar si existe mejora en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo de manera general y también en los indicadores de seguridad de confidencialidad, integridad y disponibilidad (CIA) se presenta la tabla de contingencia para la prueba chi-cuadrado, la misma que se obtuvo mediante frecuencias; representadas por la cantidad de funcionalidades vulnerables ante los ataques realizados antes y después del mantenimiento de software y al haber aplicado buenas prácticas de seguridad. Asimismo, se descartaron los ataques que no vulneraban las funcionalidades debido a que los resultados obtenidos no afectan en el cálculo del valor de la prueba estadística. Los datos obtenidos se presentan detalladamente en la **Tabla 14-4**

Tabla 14-4: Tabla de contingencia (Confidencialidad, integridad y general)

Vulnerabilidad	Funcionalidades vulneradas sin buenasprácticas (X)	Funcionalidades vulneradas con buenasprácticas (Y)
CSRF	12,00	12,00
PRIVILEGE ESCALATION	30,00	0,00
SQL Injection	12,00	0,00

Realizado por: Alex Cunachi y Erick Dávalos, 2022

- **Modelo matemático de Chi cuadrado**

Para la verificación de la hipótesis se escogió la prueba Chi Cuadrado cuya fórmula es la siguiente:

$$X^2 = \frac{\sum(O - E)^2}{E}$$

Donde:

X^2 = Chi Cuadrado

\sum = Sumatoria

O = Datos Observados

E = Datos Esperados

- **Reglas de decisión**

$$1-0,05=0,95 \quad \alpha = 0,05$$

Nivel de confiabilidad = 0,95

Nivel de significancia o error = $\alpha = 0,05$

- **Grados de Libertad**

El grado de libertad es igual a la multiplicación del número de las filas menos uno por el número de las comunas menos uno así:

$$golf = (F - 1) (C - 1)$$

$$GL = 2$$

Dónde:

GL = Grado de Libertad

C = Columnas de la Tabla

F = Filas de la Tabla

Al 95% y con 1 gl X^2_t , es igual a 5.9915

Se acepta la hipótesis nula sí, X^2_c es menor o igual a X^2_t caso contrario se rechaza con un α de 0,05

Se acepta H_0 , si c es \leq a 3,841 con α 0,05

Cálculo de chi cuadrado

Tabla 15-4: Resultados obtenidos de la prueba del chi cuadrado

O (Datos observados)	E (Datos esperados)	(O-E) ² /E
12	216/11	98/33
30	270/11	40/33
12	108/11	16/33
12	48/11	147/11
0	60/11	60/11
0	24/11	24/11
X²(calculado)		77/3 = 25.667

Realizado por: Alex Cunachi y Erick Dávalos, 2022

En la **Tabla 16-4**, se calculó el valor de chi-cuadrado es $x^2(\text{calculado}) = 25.67$ con 2 grados de libertad y un nivel de significancia del 5%, $x^2(\text{calculado}) > 5.9915$ por lo que al obtener los resultados se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_1), es decir que el uso de buenas prácticas de seguridad en el mantenimiento de software influye en la vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

Tabla 16-4: Resultados obtenidos de la prueba chi-cuadrado (Confidencialidad, integridad y general)

Prueba chi cuadrado (x^2)		
Parámetro	X	Y
Media	18	4
Varianza	72	32
Alfa	0,050	
Grados de libertad	2,000	
Estadístico x^2 (calculado)	25.667	
Estadístico x^2 (tabla)	5,9915	

Realizado por: Alex Cunachi y Erick Dávalos, 2022

En la **Ilustración 11-4**, se puede observar de manera gráfica los resultados obtenidos en la prueba chi-cuadrado después de realizar el análisis, donde se tiene la aceptación de la hipótesis H1 El uso de buenas prácticas de seguridad en el mantenimiento de software influye en la vulnerabilidad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo.

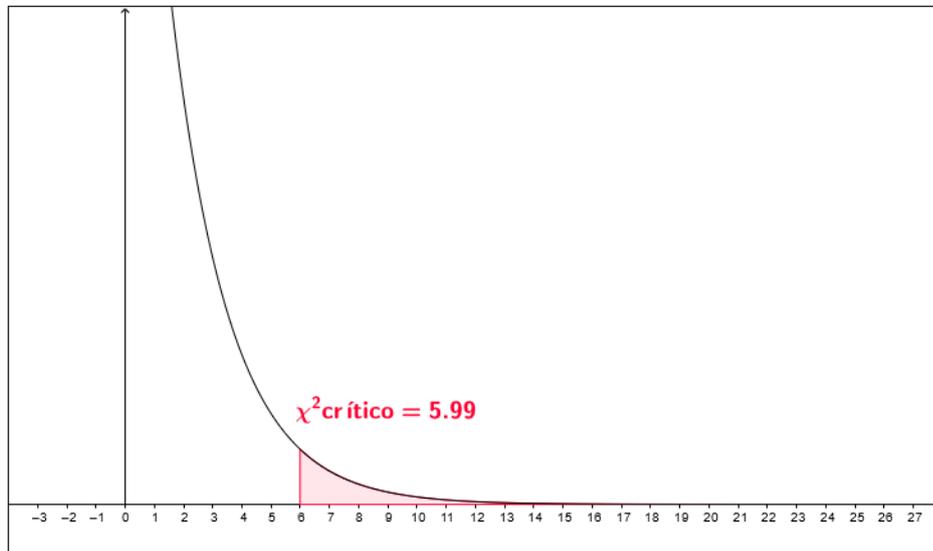


Ilustración 11-4: Distribución chi-cuadrado (Confidencialidad, integridad y general)

Realizado por: Alex Cunachi y Erick Dávalos, 2022

CONCLUSIONES

- Después de haber realizado el análisis de la arquitectura se obtuvo que el sistema de información productiva agropecuaria CONAGOPARE Chimborazo está desarrollado en la arquitectura Modelo-Vista-Controlador(MVC) en conjunto con angular para la programación del Back End y php para la programación de Front End y Mysql como gestor de base de datos.
- Mediante una investigación se determinó que la característica principal de la metodología Mantema, se enfoca en la documentación para proyectos de mantenimiento del software; además, dispone de una técnica que identifica las características de los proyectos de mantenimiento que más riesgos producen, una serie de métricas de mantenibilidad de esquemas conceptuales y lógicos de base de datos, un conjunto de métricas de tamaño y complejidad de sistemas orientadas a objeto y un conjunto de métricas para controlar el proceso de mantenimiento.
- Se creó el manual de buenas prácticas de seguridad en el cual están presentes las 10 técnicas de seguridad obtenidas de la información publicada por la fundación OWASP, para 7 ataques analizados en aplicaciones web desarrolladas en el lenguaje de programación PHP, las mismas que permiten asegurar que la información no sea manipulada o extraída de la base de datos y así evitar problemas que se puedan suscitar por inyección SQL, pérdida de autenticación, exposición de datos confidenciales, entidades externas XML, pérdida de control de acceso, secuencias de comandos entre sitios, deserialización insegura, uso de componentes vulnerables conocidos y registro y monitoreo insuficientes.
- Se realizó mantenimiento de software al sistema de información productiva agropecuaria CONAGOPARE Chimborazo, para esto se corrigieron errores por medio del mantenimiento de software de tipo correctivo, se aplicaron buenas prácticas de seguridad y se realizó mantenimiento de software de tipo perfectivo agregando 25 funcionalidades a las 32 existentes obteniendo un total de 57 requerimientos desarrollados.
- Mediante la ISO/IEC 25010 se establecieron 3 indicadores de funcionalidad, obteniendo así resultados en mejora en el sistema: para completitud funcional pasando de 22% de errores a 0%, exactitud funcional se mantuvo con 0% de errores y pertinencia funcional pasando de 22% a 0% de errores; después de haber realizado el mantenimiento de software en el sistema de información productiva CONAGOPARE Chimborazo.
- Se determinó que el nivel inicial de vulnerabilidad de seguridad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo antes de realizar el mantenimiento de software corresponde al 66.66%, al aplicar 7 ataques a las 32 funcionalidades que componen dicho sistema, en función a los parámetros CIA definidos por el estándar ISO 27001.

- El nivel final de vulnerabilidad del Sistema de información productiva agropecuaria CONAGOPARE Chimborazo después de haber realizado el mantenimiento de software y al aplicar ataques a las 57 funcionalidades desarrolladas es de 19.16%, obteniendo así una minimización de nivel de vulnerabilidad del 47,50% en cuanto a la seguridad del sistema.
- Se redujo el valor inicial de vulnerabilidad en los indicadores de CIA de seguridad después de aplicar mantenimiento software en 66,66% para el indicador de confidencialidad, 50,00% para el indicador de integridad y 25,00% para el indicador disponibilidad.

RECOMENDACIONES

- Realizar un análisis detallado de las herramientas de desarrollo, lenguajes de programación, gestores de bases de datos y metodología implementadas en los sistemas antes de aplicar un mantenimiento de software en sistema web.
- Usar del manual de buenas prácticas que se describió en este presente trabajo de integración curricular debido a que tiene como finalidad la aplicación de seguridad mediante el uso buenas prácticas de seguridad que se obtuvo especialmente para el desarrollo de aplicaciones web realizadas en el lenguaje de programación PHP en conjunto con el framework Angular.
- Investigar métodos más avanzados para aplicar pruebas de penetración a los sistemas web que permita descubrir vulnerabilidades inexistentes y por ende que no se hayan realizado un test de evaluación de seguridad, puesto que, al transcurrir el tiempo, por métodos como el hacking ético se encuentran nuevas amenazas informáticas para extracción de información de los sistemas web, y que en este presente trabajo de integración curricular no se hayan aplicado.
- Continuar con el mantenimiento de software del sistema de información productiva agropecuaria CONAGOPARE Chimborazo que permita el desarrollo de nuevas funcionalidades que el cliente vaya requiriendo en el transcurso de la vida útil del software, utilizando las actividades y tareas de la metodología Mantema, debido a que las mismas permiten realizar correctamente las tareas del proceso de mantenimiento de software del sistema.
- Investigar mejores técnicas que permita contrarrestar los ataques de tipo Cross site request forgery, debido a lo que se implementó en este trabajo de integración curricular no permitió la disminución de dicha vulnerabilidad en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo, puesto que el último registro investigado data del año 2017 y al no estar actualizado no se obtuvo un método eficaz para obtener una minimización en la vulnerabilidad evaluada.
- Actualizar el manual de buenas prácticas de seguridad con nuevas técnicas que se vayan investigando, que permitan minimizar vulnerabilidades existentes en las aplicaciones web con el objetivo de mejorar la seguridad.

GLOSARIO

Adecuación Funcional: Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. (Riascos y Rivadeneira 2015)

Amenaza: Se entiende como amenaza informática toda aquella acción que aprovecha una vulnerabilidad para atacar o invadir un sistema informático. (Ortiz 2020)

Análisis de impacto: Tiene como objetivo determinar, desde un punto de vista cuantitativo, qué elementos están realmente implicados en las peticiones de cambio solicitadas por los usuarios, una vez que los sistemas de información se encuentran en producción. (Cillero Peñuela 2009)

Pentesting: Pruebas de penetración o piratería ética, es una técnica de seguridad cibernética que las organizaciones utilizan para identificar, probar y resaltar vulnerabilidades en su postura de seguridad. Estas pruebas de penetración a menudo son realizadas por piratas informáticos éticos. (Pérez Arbesú y Osoro 2013)

Vulnerabilidad: En su sentido más amplio, el término “vulnerabilidad” está asociado con alguna violación de la política de seguridad. Esto puede ser debido a la debilidad de las reglas de seguridad, o algún problema con el propio software. En teoría, todos los sistemas informáticos tienen vulnerabilidades; su gravedad depende de que sean o no, utilizadas para causar daños al sistema. (Lagos Flores 2018)

BIBLIOGRAFÍA

AGUILERA, P., 2011. *Redes seguras (Seguridad informática)*. S.l.: Editex. ISBN 978-84-9003-112-4.

AGUIRRE, J.R., 2006. *Libro Electrónico de Seguridad Informática y Criptografía Versión 4.1*. Sexta edición. Universidad Politécnica de Madrid: s.n.

ÁLVAREZ MARAÑÓN, G., 2004. *Seguridad informática para empresas y particulares* [en línea]. España: McGraw-Hill, Interamericana de España. ISBN 978-84-481-4297-1. Disponible en: <https://books.google.com.ec/books?id=yodZJgAACAAJ>.

BALSECA RODRÍGUEZ, T.M. y CAIZA CHACÓN, G.A., 2014. ELABORACIÓN DE UNA GUÍA DE PROCEDIMIENTOS PARA EL PROCESO DE MANTENIMIENTO DE SOFTWARE. , pp. 172.

CABALLERO QUEZADA, A.E., 2021. *Hacking con Kali Linux Una Perspectiva Práctica*. , vol. Versión 3.4, pp. 121.

CALDAS HERRERA, V.A., 2015. *Desarrollo de un plan de gestión de mantenimiento de software para el Departamento de Sistemas de la Universidad Politécnica Salesiana basado en la Norma ISO/IEC 14764:2006*. Técnico. Cuenca, Ecuador: Universidad Politécnica Salesiana.

CILLERO PEÑUELA, M., 2009. Análisis de Impacto. *manuel.cillero.es* [en línea]. [Consulta: 11 octubre 2022]. Disponible en: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/analisis-de-impacto/>.

COBO, Á., GÓMEZ, P., PÉREZ, D. y ROCHA, R., 2005. *PHP y MySQL: tecnologías para el desarrollo de aplicaciones web*. España: Ediciones Daz de Santos. ISBN 978-84-7978-706-6.

CUEVAS CALLISAYA, B.P., 2006. *Modelo de Ingeniería del Mantenimiento para Productos Software Orientado a Objetos*. La Paz – Bolivia: UNIVERSIDAD MAYOR DE SAN ANDRES. Carrera de Informática

EGUILUZ, J., 2008. *Introducción a JavaScript*., pp. 134.

FERNÁNDEZ, L., 2020. OWASP Top Ten: Proyecto clave para la seguridad de las aplicaciones. *RedesZone* [en línea]. Disponible en: <https://www.redeszone.net/tutoriales/seguridad/que-es-owasp-seguridad-aplicaciones/>.

GOMEZ VIEITES, A., 2006. *Enciclopedia de la Seguridad Informática*. (Madrid, España): RA-MA S.A. Editorial y Publicaciones.

IEEE, 1993. *ESTÁNDAR IEEE 1219 MANTENIMIENTO DEL SOFTWARE*. S.l.: s.n.

ISO/IEC 25000:2014, 2014. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*. ISO [en línea]. [Consulta: 8 agosto 2022]. Disponible en: <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/47/64764.html>.

KALI.ORG, 2022. What is Kali Linux? | Kali Linux Documentation. *Kali Linux* [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://www.kali.org/docs/introduction/what-is-kali-linux/>.

LAGOS FLORES, E., 2018. *Análisis de vulnerabilidades y pruebas de penetración a la infraestructura tecnológica de empresas*. Ciudad Universitaria, Cd. Mx.: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

LÓPEZ MARTÍNEZ, J.A., 2017. *Adaptación del proceso de desarrollo software para cumplimiento de la adecuación funcional según ISO/IEC 25000*. Técnico. Ciudad Real – Spain: UNIVERSIDAD DE CASTILLA-LA MANCHA.

MONTORO, M., 2020. ¿Qué son las buenas prácticas y para qué sirven? *ATS* [en línea]. [Consulta: 12 mayo 2022]. Disponible en: <https://www.ats.edu.uy/buenas-practicas/>.

MOQUILLAZA HENRÍQUEZ, S.D., VEGA HUERTA, H. y GUERRA GRADOS, L., 2010. Programación en N capas. , pp. 11.

ORDOÑEZ GRANIZO, S.J. y CHIMBO ORTIZ, K.D., 2019. *BUENAS PRÁCTICAS DE SEGURIDAD EN EL SISTEMA DE ESTAFETAS DE LA DIRECCIÓN DE DESARROLLO ACADÉMICO EN LA ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO*. Proyecto Técnico. Riobamba: ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

ORTIZ, A.E., 2020. ¿Qué es una amenaza informática? ¿Cómo contenerla? [en línea]. [Consulta: 11 octubre 2022]. Disponible en: <https://www.hostdime.la/blog/que-es-una-amenaza-informatica-como-contenerla/>.

PASQUALI, S., 2013. *Mastering Node.js*. Packt Publishing Limited. United Kingdom: s.n.

PÉREZ ARBESÚ, L. y OSORES, M., 2013. ¿Qué es Prueba de penetración (pen test)? - Definición en WhatIs.com. *ComputerWeekly.es* [en línea]. [Consulta: 11 octubre 2022]. Disponible en: <https://www.computerweekly.com/es/definicion/Prueba-de-penetracion-pen-test>.

PIATTINI VELTHUIS, M.G., POLO USAOLA, M., RUIZ GONZÁLEZ, F., GARCIA RODRIGUEZ DE GUZMAN, I. y PÉREZ DEL CASTILLO, R., 2018. *Mantenimiento y Evolución de Sistemas de información*. 1. España: RA-MA S.A. Editorial y Publicaciones. ISBN 978-84-9964-759-3.

PRESSMAN, R.S., 1998. *INGENIERÍA DEL SOFTWARE UN ENFOQUE PRACTICO*. Cuarta edición. España: McGrawhill.

RIASCOS, G. y RIVADENEIRA, D., 2015. Funcionalidad · Atributos de calidad y patrones de diseño. [en línea]. [Consulta: 27 abril 2022]. Disponible en: <https://fjimenezg.gitbooks.io/atributos-de-calidad-y-patrones-de-diseno/content/funcionalidad.html>.

RIVERA, D., 2019. Qué es OWASP y por qué todo desarrollador debería conocerlo. [en línea]. [Consulta: 27 abril 2022]. Disponible en: <https://blog.pleets.org/article/conoce-owasp>.

ROBLES, V., 2017. ¿Qué es Angular y para qué sirve? *Victor Robles* [en línea]. [Consulta: 13 octubre 2022]. Disponible en: <https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>.

ROMERO CASTRO, M.I., FIGUEROA MORÁN, G.L., VERA NAVARRETE, D.S., ÁLAVA CRUZATY, J.E., PARRALES ANZÚLES, G.R., ÁLAVA MERO, C.J., MURILLO QUIMIZ, Á.L. y CASTILLO MERINO, M.A., 2018. *Introducción a la seguridad informática y el análisis de vulnerabilidades* [en línea]. 1. S.l.: Editorial Científica 3Ciencias. [Consulta: 14 enero 2021]. ISBN 978-84-949306-1-4. Disponible en:

<https://www.3ciencias.com/libros/libro/introduccion-a-la-seguridad-informatica-y-el-analisis-de-vulnerabilidades/>.

RUIZ, F. y POLO, M., 2001. *Mantenimiento del Software.*, pp. 31.

SANTOS, Y., 2015. *Introducción a las pruebas de penetración.*,

SILBERSCHATZ, A., KORTH, H.F. y SUDARSHAN., S., 2002. *Fundamentos de bases de datos.* 4a. ed. Madrid: McGraw-Hill: s.n.

VOUTSSAS, 2010. *Preservación documental digital y seguridad informática.* [en línea].

[Consulta: 13 octubre 2022]. Disponible en:

https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-358X2010000100008&lng=es&nrm=iso.

Anexo A: Proceso de mantenimiento de software: Mantema



PROCESO DE MANTENIMIENTO DE SOFTWARE: MANTEMA

AUTORES:

ERICK FABRICIO DÁVALOS MANCERO

ALEX SANTIAGO CUNACHI CHACHA

Riobamba – Ecuador

2022

Tabla de contenidos

Introducción del sistema.....	1
1. Sección 1: Solicitud de modificación.....	2
2. Sección 2: Planificación del proceso de mantenimiento de software	4
3. Sección 3: Inventario de documentación del sistema desarrollado.....	6
4. Sección 4: Tipo de modificación a realizarse en el sistema.....	7
5. Sección 5: Proceso de gestión de riesgos.....	8
5.1. Identificación de riesgos.....	8
5.2. Análisis y priorización del riesgo	8
5.3. Plan de reducción, supervisión y gestión del riesgo	10
6. Sección 6: Estudio de factibilidad	20
6.1. Factibilidad técnica.....	20
6.2. Factibilidad operativa.....	21
6.3. Factibilidad económica.....	22
7. Sección 7: Solución al mantenimiento de software.....	23
8. Sección 8: Modelo entidad-relación y modelo físico de la base de datos desarrollada del sistema antiguo	24
9. Sección 9: Historias de usuario	26
10. Sección 10: Resultados de las pruebas unitarias y de integración en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo.....	51
11. Sección 11: Sistema de información productiva CONAGOPARE Chimborazo antes y después del mantenimiento de software	52
12. Sección 12: Modelo conceptual y físico de la base de datos actualizada después de aplicar mantenimiento de software.....	54
13. Sección 13: Diccionario de datos de la base de datos actualizada después del mantenimiento de software.....	56

Introducción del sistema

El sistema al cual se va a dar mantenimiento se denomina Sistema de información productiva agropecuaria CONAGOPARE Chimborazo, que permite el ingreso de información productiva agropecuaria de las parroquias que componen la provincia de Chimborazo, lo que ayuda a llevar un buen registro por parroquia de la información tanto de agricultura como pecuaria.

La información es registrada por parroquia y se puede realizar impresiones tanto de impresiones como también por escrito, además de una gestión de documento en donde pasa todo documento a ser revisado y autorizado por el presidente de la Institución.

1. Sección 1: Solicitud de modificación

Tabla A.1: Formato de solicitud de modificación

Fecha	04/04/2022	Id solicitud	CN001
Tipo de solicitud			
Nombre del proyecto	Mantenimiento del sistema de información productiva CONAGOPARE Chimborazo		
Líder del proyecto	Ing. Marco Álvarez		
Descripción de la solicitud	Aumento de funcionalidades en el sistema Implementación de seguridad Actualización de documentación		
Solicitado por			

Requerimientos

N°	Descripción	Tipo de mantenimiento	Prioridad
RF01	Registro de una empresa	Perfectivo	Alta
RF02	Registro de una empresa productiva		
RF03	Modificar de una empresa productiva		
RF04	eliminar una empresa productiva		
RF05	registrar un usuario		
RF06	modificar un usuario		
RF07	eliminar un usuario		
RF08	el registro de movilidad		
RF09	modificar movilidad		
RF10	eliminar movilidad		
RF11	el registro de económico social		
RF12	modificar económico social		
RF13	eliminar económico social		
RF14	generar una solicitud		
RF15	asignar un estado a una solicitud		
RF16	generar memorándums para asignación de técnicos a proyectos		
RF17	El sistema emitirá un listado de empresas productivas registradas en el sistema		

N°	Descripción	Tipo de mantenimiento	Prioridad
RF18	El sistema emitirá un listado de registros de movilidad ingresados en el sistema	Perfectivo	Alta
RF19	El sistema emitirá un listado de registros de económico social ingresados en el sistema		
RF20	El sistema emitirá un listado los usuarios registrados en el sistema		
RF21	El sistema emitirá un listado de las todas solicitudes para el administrador		
RF22	El sistema mostrará usuario según su nombre, parroquia		
RF23	El sistema mostrará las empresas productivas según su nombre, cantidad de empleados		
RF24	El sistema mostrará los registros de movilidad según su estado, mantenimiento, kilometraje		
RF25	El sistema emitirá un listado de todas las solicitudes aprobadas según su parroquia, rangos de fechas, técnicos asignados		

Resultado de la solicitud

Estado de la solicitud	En proceso	
Autorización	Revisado por	
	Gestor de peticiones	

2. Sección 2: Planificación del proceso de mantenimiento de software

Tabla B.2: Planificación (Plantilla 2)

Datos generales del sistema										
Nombre del sistema		Sistema de información productiva agropecuaria CONAGOPARE Chimborazo								
Líder del sistema		Marco Álvarez								
Responsable del mantenimiento		Alex Cunachi, Erick Dávalos								
Planificación										
Periodos						Responsables				
Id. Solicitud	Período 1	Período 2	Período 3	Período 4	Fin del proyecto	Equipo de mantenimiento				
CN001	04/04/2022 15/04/2022	18/04/2022 29/04/2022	02/05/2022 27/05/2022	30/05/2022 30/06/2022	01/07/2022 20/07/2022	Responsable del mantenimiento	Analista programador	SQA	Tester	Documentador
Revisión y análisis de la documentación actual y aprobación del mantenimiento						X				
Investigación sobre metodologías para mantenimiento de software						X				

Investigación y aplicación de buenas prácticas de seguridad						X				
Desarrollo de los módulos empresas productivas y movilidad						X				
Desarrollo de los módulos económico social y gestión documental						X		X		
Pruebas al sistema después de haber realizado el mantenimiento									X	
Actualización de documentos										X

3. Sección 3: Inventario de documentación del sistema desarrollado

Tabla B.3: Inventario de la documentación del sistema actual

Datos generales del sistema					
Nombre del sistema		Sistema de información productiva agropecuaria CONAGOPARE Chimborazo			
Líder del sistema		Marco Álvarez			
Responsable del mantenimiento		Alex Cunachi, Erick Dávalos			
Inventario de la documentación del sistema actual					
Fases de desarrollo del sistema	N°	Id-documento	Nombre del documento	Versión	Observaciones
Gestión de requerimientos	1	Conago_Ges_Fun	Gestión de requerimientos	V1.0	
	2	Conago_Ges_Ries	Plan de gestión de riesgos	V1.0	
Planificación del proyecto	3	Conago_Pln_Adm	Plan de administración del proyecto	V1.0	
	4	Conago_Mol_Bases	Modelo de bases de datos	V1.0	
Análisis y diseño	5	Conago_Cod_Progra	Códigos de programación del sistema	V1.0	
	6	Conago_Man_Tec	Manual técnico	V1.0	
	7	Conago_Man_Usu	Manual de usuario	V1.0	
Implementación	8	Doc_final	Documento del sistema final	V1.0	
	9	Sis_Info_Prod_Agro	Sistema de información productiva CONAGOPARE Chimborazo	V1.0	
Administración del proyecto	10	Conago_Acta_Segu	Acta de seguimiento interno	V1.0	

4. Sección 4: Tipo de modificación a realizarse en el sistema

Tabla B.4: Decisión tipo de modificación

Datos generales del sistema										
Nombre del sistema		Sistema de información productiva agropecuaria Conagopare Chimborazo								
Líder del sistema		Marco Álvarez								
Solicitud de modificación										
Id. Solicitud	Descripción o destalle	Responsable del mantenimiento	Fecha recepción	Fecha aprobación	Fecha cierre	Tipo de mantenimiento	Motivo de selección	Prioridad	Estado del cambio	Estado de implementación
	Desarrollo módulo de empresas productivas	Alex Cunachi Erick Dávalos	04/04/2022	15/04/2022		Perfectivo Correctivo	Se realizó la selección de este tipo de mantenimiento debido a que se va a implementar funcionalidades nuevas en el sistema	Alta	Aprobado	No iniciado
	Desarrollo de movilidad									
	Desarrollo módulo económico									
	Desarrollo módulo gestión documental									
	Aplicación de buenas prácticas de seguridad									

5. Sección 5: Proceso de gestión de riesgos

5.1. Identificación de riesgos

Tabla A.5: Identificación del Riesgo

Identificación	Descripción del riesgo	Tipo de Riesgo	Consecuencias
R1	Mal diseño de la Base de Datos	R. Proyecto	Redundancia e inconsistencia de datos.
R2	Interfaces poco amigables con el usuario.	R. Proyecto	Dificultad en el manejo del sistema
R3	Mala recolección de información para los requisitos funcionales.	R. Proyecto	Al no tener claro el problema de la empresa y la solución que se va a dar a este, se ocasiona un retraso y altos costos de rectificación en el proyecto.
R4	Mala planificación y estimación de tiempo del proyecto	R. Proyecto	Incremento de costo y tiempo en la entrega del proyecto
R5	Robo o daño en el equipo informático usado para el desarrollo del sistema.	R. Técnico	Incremento en el costo, retraso en la entrega de los módulos y pérdida de información
R6	Cambio de Requisitos	R. Técnico	Retraso de la entrega de los módulos en el proyecto.
R7	Cambio de directivos y representantes en CONAGOPARE Chimborazo	R. Negocio	Suspensión parcial o definitiva del proyecto
R8	Cambio en las políticas de la empresa	R. Negocio	Rediseño completo o parcial del Software
R9	Mala identificación de las herramientas, arquitectura y lenguaje usado en la creación de la aplicación	R. Proyecto	Si no se tiene claro cómo se construyó la aplicación el mantenimiento conllevará más tiempo de lo estimado

5.2. Análisis y priorización del riesgo

En esta sección se analizarán las características de los riesgos, con sus respectivos parámetros y rangos de probabilidad, que puedan presentarse mientras se desarrolla el proyecto de integración curricular.

La presente tabla describe los posibles problemas que se puedan presentar y que el equipo de desarrollo ha tomado en cuenta, haciendo representativa su especificación.

Tabla A.6: Análisis del riesgo

Identificación	Probabilidad			Impacto		Exposición al riesgo	
	%	Valor	Probabilidad	Valor	Impacto	Valor	Exposición
R1	37%	2	Media	4	Crítico	8	Alta
R2	39%	2	Media	2	Moderado	4	Media
R3	78%	3	Alta	4	Crítico	12	Crítico
R4	30%	1	Bajo	2	Moderado	2	Bajo
R5	32%	1	Bajo	3	Alto	3	Medio
R6	55%	2	Media	3	Alto	6	Alta
R7	83%	3	Alta	4	Crítico	12	Alta
R8	70%	3	Alto	4	Crítico	12	Alta
R9	65%	2	Media	3	Alto	6	Media

Tabla A.7: Priorización del riesgo

Identificación	Descripción	Exposición	Valor	Prioridad
R7	Cambio de directivos y representantes en Conagopare Chimborazo	Alto	12	1
R8	Cambio en las políticas de la empresa	Alto	12	1
R3	Mala recolección de información para los requisitos funcionales.	Alto	12	1
R6	Cambio de Requisitos	Alto	8	2
R1	Mal diseño de la Base de Datos	Alto	6	3
R2	Interfaces poco amigables con el usuario.	Medio	4	4
R5	Robo o daño en el equipo informático usado para el desarrollo del sistema.	Medio	3	5
R9	Mala identificación de las herramientas, arquitectura y lenguaje usado en la creación de la aplicación.	Medio	6	6
R4	Mala planificación y estimación de tiempo del proyecto	Bajo	2	6

5.3. Plan de reducción, supervisión y gestión del riesgo

Tabla A.8: Plan de Reducción, Supervisión y Gestión del Riesgo

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R1		FECHA: 10/5/2022	
Probabilidad: Media Valor: 2	Impacto: Crítico Valor: 4	Exposición: Alto Valor: 8	Prioridad: 3
DESCRIPCIÓN: Mal diseño de la Base de Datos			
REFINAMIENTO:			
<u>Causas:</u>			
<ul style="list-style-type: none"> • Mala comunicación entre cliente y el desarrollador del proyecto • Falta de atributos y tablas en la base de datos • Mala relación entre tablas 			
<u>Consecuencias:</u>			
<ul style="list-style-type: none"> • Retraso del proyecto • No cumplir con los requisitos establecidos 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Establecer las políticas, procedimientos y documentación que es necesario recopilar para la planificación, ejecución y control de la programación del proyecto. • Identificar y documentar las acciones concretas que será necesario realizar para producir los entregables del proyecto. • Definir las relaciones entre las distintas actividades del proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control de cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Aplicar el cambio lo más rápido posible. • Recomposición del esquema. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input type="checkbox"/>	
Gestionando el riesgo:		<input checked="" type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R2		FECHA: 10/5/2022	
Probabilidad: Media Valor: 2	Impacto: Moderado Valor: 2	Exposición: Media Valor: 4	Prioridad: 4
DESCRIPCIÓN: Interfaces poco amigables con el usuario			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Mala comunicación entre cliente y el desarrollador del proyecto • Mala elección de colores, formas y ubicación de los elementos en pantalla 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso del proyecto • Desorientación del cliente 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Establecer las políticas, procedimientos y documentación que es necesario recopilar para la planificación, ejecución y control de la programación del proyecto. • Identificar y documentar las acciones concretas que será necesario realizar para producir los entregables del proyecto. • Definir las relaciones entre las distintas actividades del proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control de cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Aplicar el cambio lo más rápido posible. • Recomposición del esquema. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha, Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R3		FECHA: 10/5/2022	
Probabilidad: Alta Valor: 3	Impacto: Crítico Valor: 4	Exposición: Alto Valor: 12	Prioridad: 1
DESCRIPCIÓN: Mala recolección de información para los requisitos funcionales			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Mala comunicación entre cliente y el desarrollador del proyecto • Dificultad del cliente para relacionar sus necesidades con los requerimientos dados • Dificultad del desarrollador de capturar la información relevante de los requisitos 			
Consecuencias:			
<ul style="list-style-type: none"> • Incremento en los costos de desarrollo • Retraso del proyecto • Difícil mantenimiento del software 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Interacción con el cliente en cada fase del desarrollo para ir validando los requerimientos • Documentar cada requisito e ir controlando el cumplimiento de los mismos. • Si es posible corregir los errores antes de que inicie el proyecto 			
SUPERVISION:			
<ul style="list-style-type: none"> • Grado de compromiso del desarrollador en el proyecto • Mejor relación del desarrollador con el cliente • Verificar la correcta adaptación de los nuevos cambios al proyecto 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Flexibilidad adaptando los nuevos cambios sin afectar los avances desarrollados • Estimar nuevos costos por los cambios a realizar • Realizar cambios con el menor costo • Llegar a un acuerdo con el cliente sobre el incremento del costo y la fecha de entrega del proyecto por los nuevos cambios a realizar. • Nueva asignación de recursos y reajuste de planificación 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input type="checkbox"/>	
Gestionando el riesgo:		<input checked="" type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R4		FECHA: 10/5/2022	
Probabilidad: Baja Valor: 1	Impacto: Moderado Valor: 2	Exposición: Baja Valor: 2	Prioridad: 6
DESCRIPCIÓN: Mala planificación y estimación de tiempo del proyecto			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Falta de comunicación con el cliente • Visión del desarrollador diferente que la del cliente • Dificultad del desarrollador de capturar la información relevante de los requisitos 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso del proyecto • Difícil mantenimiento del software 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Interacción con el cliente en cada fase del desarrollo para ir validando los requerimientos • Documentar cada requisito e ir controlando el cumplimiento de los mismos. • Si es posible corregir los errores antes de que inicie el proyecto 			
SUPERVISION:			
<ul style="list-style-type: none"> • Grado de compromiso del desarrollador en el proyecto • Mejor relación del desarrollador con el cliente • Verificar la correcta adaptación de los nuevos cambios al proyecto 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Reajuste de planificación • Estimar nuevos costos por los cambios a realizar • Realizar cambios con el menor costo • Llegar a un acuerdo con el cliente sobre el incremento del costo y la fecha de entrega del proyecto por los nuevos cambios a realizar. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input type="checkbox"/>	
Gestionando el riesgo:		<input checked="" type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha, Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R5		FECHA: 10/5/2022	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Medio Valor: 3	Prioridad: 5
DESCRIPCIÓN: Robo o daño en el equipo informático usado para el desarrollo del sistema.			
REFINAMIENTO:			
<u>Causas:</u>			
<ul style="list-style-type: none"> • Equipos antiguos de baja calidad • Lugar de trabajo inadecuado • Sobrecarga de trabajo en equipos de poca fidelidad 			
<u>Consecuencias:</u>			
<ul style="list-style-type: none"> • Incremento en los costos de desarrollo • Retraso del proyecto • Difícil mantenimiento del software 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Adquisición de nuevos dispositivos para un desempeño óptimo • Si es posible corregir los errores antes de que inicie el proyecto • Elegir una metodología de desarrollo que sea flexible a cambios 			
SUPERVISION:			
<ul style="list-style-type: none"> • Grado de compromiso del desarrollador en el proyecto • Mejor relación del desarrollador con el cliente • Verificar la correcta adaptación de los nuevos cambios al proyecto 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Estimar nuevos costos por los cambios a realizar • Llegar a un acuerdo con el cliente sobre el incremento del costo y la fecha de entrega del proyecto por los nuevos cambios a realizar. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R6		FECHA: 10/5/2022	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alto Valor: 6	Prioridad: 2
DESCRIPCIÓN: Cambios de Requisitos			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Mala planificación del tiempo de desarrollo. • Mala comunicación con el cliente. 			
Consecuencias:			
<ul style="list-style-type: none"> • Reducción de ganancias. • Pérdida de tiempo. 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Establecer las políticas, procedimientos y documentación que es necesario recopilar para la planificación, ejecución y control de la programación del proyecto. • Identificar y documentar las acciones concretas que será necesario realizar para producir los entregables del proyecto. • Definir las relaciones entre las distintas actividades del proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control del cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Reajuste de planificación • Estimar nuevos costos por los cambios a realizar • Llegar a un acuerdo con el cliente sobre el incremento del costo y la fecha de entrega del proyecto por los nuevos cambios a realizar. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha, Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R7		FECHA: 10/5/2022	
Probabilidad: Alto Valor: 3	Impacto: Crítico Valor: 4	Exposición: Alto Valor: 12	Prioridad: 1
DESCRIPCIÓN: Cambio de directivos y representantes en Conagopare Chimborazo			
REFINAMIENTO:			
<u>Causas:</u>			
<ul style="list-style-type: none"> • Falta de presupuesto para el desarrollo del proyecto. • No realizar entregas de avances del proyecto en fechas ya determinadas. • Mal ambiente de trabajo • Malos entendidos entre directivos y programadores 			
<u>Consecuencias:</u>			
<ul style="list-style-type: none"> • Replanificación de tareas dadas por la nueva directiva. • Suspensión definitiva del proyecto • Retraso en la presentación del proyecto final. 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Aceptar los cambios planteados en reuniones con los clientes. • Reunirse con el personal para determinar las causas del cambio de Directiva del proyecto. • Actuar para reducir estas causas antes de que continúe el proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control del cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Reajuste de planificación • Realizar reuniones diarias con los nuevos Clientes. • Aceptar los nuevos cambios de funcionalidades que poseerá el sistema. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R8		FECHA: 10/5/2022	
Probabilidad: Alto Valor: 3	Impacto: Crítico Valor: 4	Exposición: Alto Valor: 12	Prioridad: 1
DESCRIPCIÓN: Cambio en las políticas de la empresa			
REFINAMIENTO:			
<u>Causas:</u>			
<ul style="list-style-type: none"> • Falta de presupuesto para el desarrollo del proyecto. • No realizar entregas de avances del proyecto en fechas ya determinadas. • Mal ambiente de trabajo • Malos entendidos entre directivos y programadores 			
<u>Consecuencias:</u>			
<ul style="list-style-type: none"> • Replanificación de tareas dadas por la nueva directiva. • Suspensión definitiva del proyecto • Retraso en la presentación del proyecto final. 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Aceptar los cambios planteados en reuniones con los clientes. • Reunirse con el personal para determinar las causas del cambio de Directiva del proyecto. • Actuar para reducir estas causas antes de que continúe el proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control del cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Reajuste de planificación • Aceptar los nuevos cambios en las políticas de la empresa • Aceptar los nuevos cambios de funcionalidades que poseerá el sistema. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

HOJA DE INFORMACION DEL RIESGO			
ID. DEL RIESGO: R9		FECHA: 10/5/2022	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Media Valor: 6	Prioridad: 6
DESCRIPCIÓN: Mala identificación de las herramientas, arquitectura y lenguaje usado en la creación de la aplicación			
REFINAMIENTO:			
<u>Causas:</u>			
<ul style="list-style-type: none"> • No identificar la arquitectura con la cual fue desarrollada la aplicación. • No identificar las herramientas y el lenguaje utilizado. 			
<u>Consecuencias:</u>			
<ul style="list-style-type: none"> • Pérdida de tiempo al tratar de identificar las herramientas y lenguaje. • Si no se identifica la arquitectura que fue desarrollada la aplicación puede ocasionar que la conexión entre las capas no se pueda realizar 			
ACCIONES DE REDUCCIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Aceptar los cambios planteados en reuniones con los clientes. • Reunirse con el personal para determinar las causas del cambio de Directiva del proyecto. • Actuar para reducir estas causas antes de que continúe el proyecto. 			
SUPERVISION:			
<ul style="list-style-type: none"> • Desarrollar el cronograma de proyecto. • Control del cronograma. • Actualizar el avance del proyecto. 			
ACCIONES GESTIÓN DEL RIESGO:			
<ul style="list-style-type: none"> • Reajuste de planificación • Aceptar los nuevos cambios en las políticas de la empresa • Aceptar los nuevos cambios de funcionalidades que poseerá el sistema. 			
ESTADO ACTUAL:			
Fase de reducción iniciada		<input type="checkbox"/>	
Fase de Supervisión iniciada		<input checked="" type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Alex Santiago Cunachi Chacha			
Erick Fabricio Dávalos Mancero			

Criterios De Valoración

Criterios de valoración de la probabilidad

La probabilidad de que ocurra un riesgo ha sido cuantificada de acuerdo a los siguientes criterios:

RANGO DE PROBABILIDADES	DESCRIPCIÓN	VALOR
1% - 33%	BAJA	1
34% – 67%	MEDIA	2
68% -99%	ALTA	3

Criterios de valoración del impacto

El impacto del riesgo ha sido valorado en función de aspectos como retrasos en la entrega del producto e impacto técnico de acuerdo a los siguientes parámetros:

IMPACTO	RETRASO	IMPACTO TECNICO	COSTO	VALOR
BAJO	1 semana	Ligero efecto en el desarrollo del proyecto	< 1%	1
MODERADO	2 semanas	Moderado efecto en el desarrollo del proyecto	<5%	2
ALTO	1 mes	Severo efecto en el desarrollo del proyecto	<10%	3
CRÍTICO	Más de un mes	Proyecto no puede ser culminado	>10%	4

Criterios de valoración de la exposición al riesgo

La exposición al riesgo ha sido determinada multiplicando la probabilidad del riesgo y el impacto del riesgo y se la ha categorizado de la siguiente manera:

EXPOSICIÓN AL RIESGO	VALOR	COLOR
BAJA	1 o 2	1
MEDIA	3 o 4	2
ALTA	Mayor a 6	3

Impacto Probabilidad.	BAJO = 1	MODERADO= 2	ALTO =3	CRITICO=4
ALTA = 3	3	6	9	12
MEDIA= 2	2	4	6	8
BAJA = 1	1	2	3	4

6. Sección 6: Estudio de factibilidad

6.1. Factibilidad técnica

- **Hardware existente**

Cantidad	Descripción	Estado
1	Laptop Asus 64 bits, procesador core i7, memoria ram 8Gb	Muy Bueno
1	Laptop Dell 64 bits, procesador core i5, memoria ram 8Gb	Muy Bueno
1	Router	Muy Bueno
1	Switchs	Muy Bueno

- **Hardware requerido**

Cantidad	Descripción	Observaciones
1	Servidor Apache	Para almacenar las API del sistema de producción agropecuaria Conagopare Chimborazo y la aplicación del sistema además debe contar con una base de datos de preferencia MySql
1	Impresora	Para imprimir documentos requeridos para la construcción del sistema.

- **Software**

- **Software existente**

Nombre	Descripción	Estado
Sistema operativo	Windows 10	No Legal
Sistema gestor de base de datos	MySQL	Legal
Entorno de desarrollo integrado (IDE)	Visual Studio Code	Legal
Herramienta de Gestión de pruebas	TestLink	Legal
Gestor de Referencias Bibliográficas	Zotero	Legal
Plataforma de desarrollo colaborativo	GitHub	Legal

- **Software requerido**

Nombre	Descripción	No. Licencias
Sistema operativo	Windows 10	2
Antivirus	Kaspersky Internet Security	2
Template para diseño web	Angular Admin Template COREui PRO	2

- **Personal técnico**
- **Personal técnico existente**

Nombre	Descripción
Alex Santiago Cunachi Chacha	Analista, diseñador, programador y tester
Erick Fabricio Dávalos Mancero	Analista, diseñador, programador y tester

- **Personal técnico requerido**

Función	Formación Académica	Experiencia en
Ing. Diseño Gráfico	Tercer Nivel	Ambientes de proyectos de sistemas informáticos
Ing. Comercial	Tercer Nivel	Líder Comercial, Compras y Ventas
Ing. Contabilidad y Auditoría	Tercer Nivel	Líder Financiera
Ing. Administración de Empresas	Tercer Nivel	Administración de Empresas

6.2. Factibilidad operativa

Tipo	Requerimiento	Estado	Descripción
Hardware	Servidor	En funcionamiento	Se cuenta con un servidor de base de datos para almacenar la información del sistema web y las APIs
Hardware/ Software	Accesibilidad	En funcionamiento	El sitio es accesible desde cualquier parte del mundo y en cualquier dispositivo con una conexión a internet.
Hardware/ Software	Disponibilidad	En funcionamiento	El sitio web está operativo las 24 horas los 7 días de la semana.
Software	Asignación de usuario y password	En funcionamiento	La aplicación web asigna un usuario y un password a los técnicos de las 45 parroquias de Chimborazo

6.3. Factibilidad económica

Duración del proyecto integrador = 4 meses

- **Costos de desarrollo**

Costos de Personal				
	Cant. Personal	Meses	Mensual	Total
Costos de analista	1	1	800	800
Costo de programadores	2	3	700	4200
Costo de diseñadores	2	1	700	1400
Costos de tester	2	1	1200	2400
Costo total de personal				8.800

Costos de Hardware y Software	
Descripción	Costo(mes)
Costos del servidor (Hosting Xpress 500 Mb, Dominios: com - info)	2.25
Costos de Suministros	50
Costos de Capacitación al personal	28.57
Costos de software de desarrollo	822.12
Otros costos (internet, luz, entre otros)	69.31
Total, costo de desarrollo del sistema mensual	971.95

- **Costos de instalar el sistema**

Costos de instalar el sistema	
Descripción	Costo
Costos de capacitación a usuarios	30
Costo de personal durante la instalación	100
Costo de instalación comercial	
Hosting	15
Dominio	12
Costo de ejecuciones en paralelo	100
Costo de adecuación de local	30
Costo red de comunicaciones	22.31
Total, costo de instalación del sistema	309.31

- **Costos de operación**

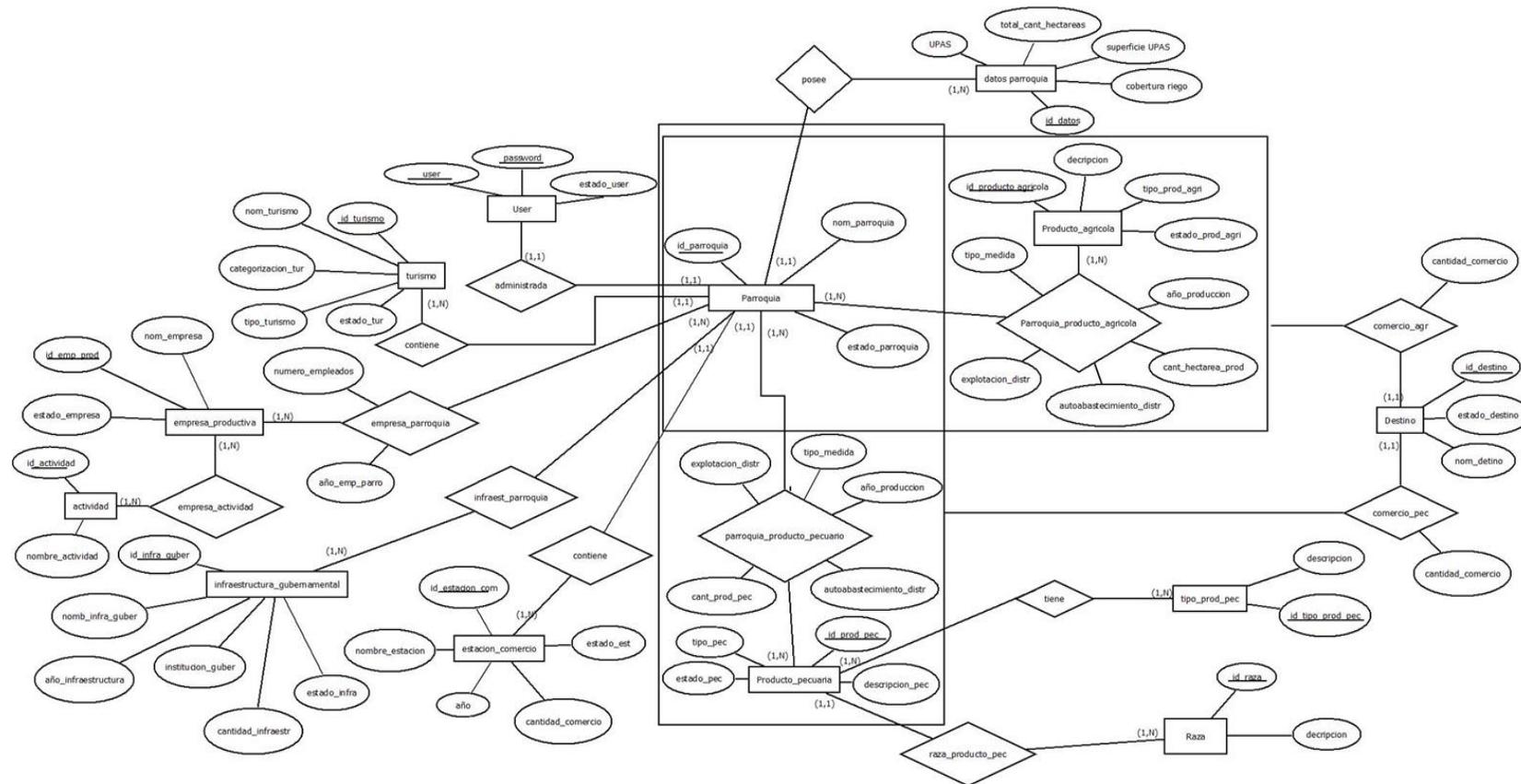
Costos de operación		
Descripción	Costo unitario	Costo total
Mensualmente		902.31
Costos de personal de Operación	800	
Costos de operación de red comunicaciones	22.31	
Costos de materiales y suministros Otros costos	80	
Anualmente		827
Costo mantenimiento sistema web	800	
Costo operación de computadora central (servidor)	27	
Total, costos de operación anualmente		11654.72

7. Sección 7: Solución al mantenimiento de software

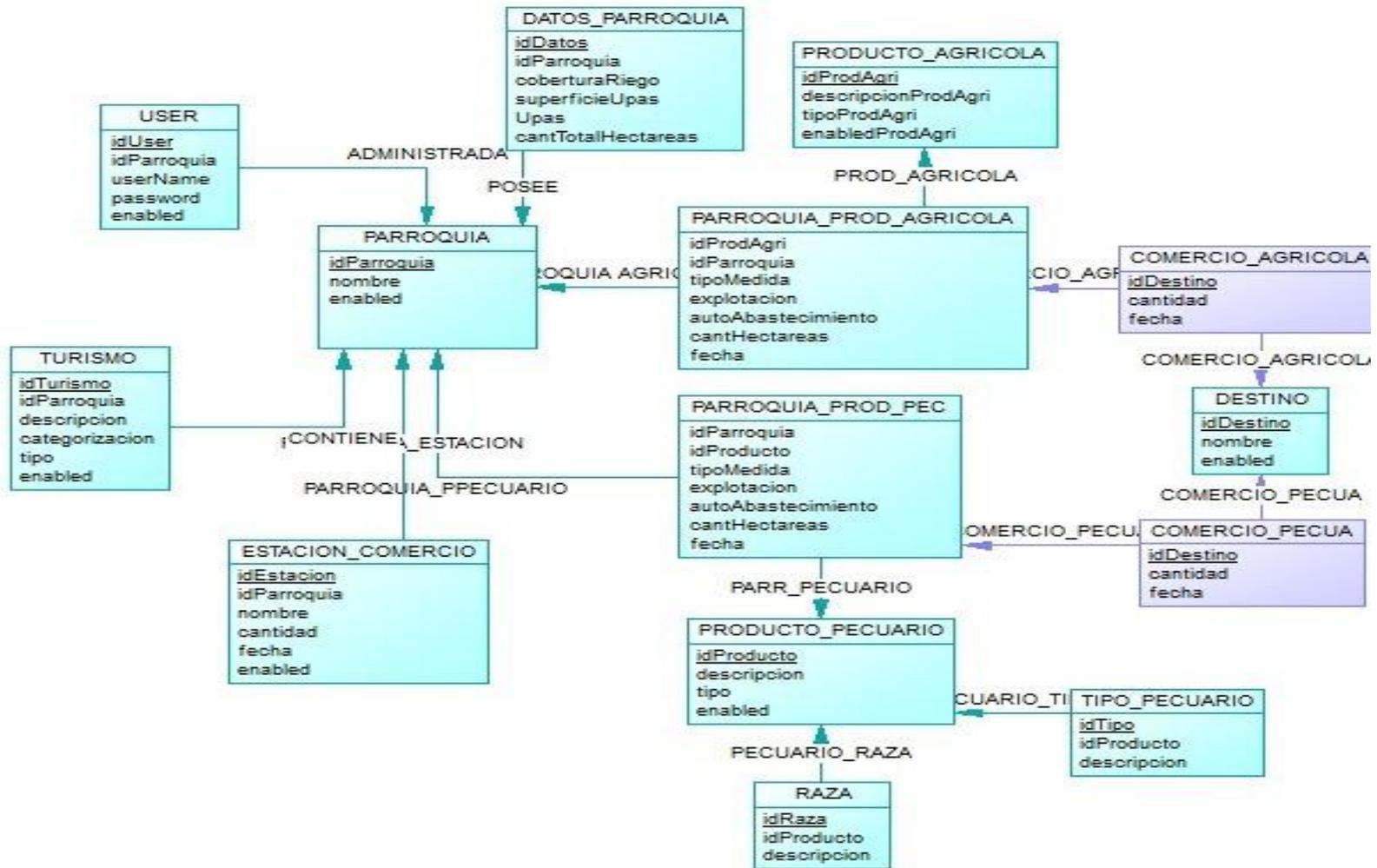
Datos generales del sistema						
Nombre del sistema	Sistema de información productiva agropecuaria CONAGOPARE Chimborazo					
Líder del sistema	Marco Álvarez					
Responsable del mantenimiento	Alex Cunachi, Erick Dávalos					
Estado de elementos a modificar						
N°	Elemento a modificar	Capa de la arquitectura	Descripción del cambio a implementar	Responsable de la modificación	Duración Estado de la modificación	Observaciones
1	Sistema	Vista	Crear elementos de la interfaz de usuario Implementar pantallas para cada módulo a desarrollar Realizar respando los estándares de desarrollo.	Alex Cunachi Erick Dávalos	4 meses	
2		Controlador	Crear un control para cada reporte, implementar las acciones que devulven los datos para la vista	Alex Cunachi Erick Dávalos		
3		Gestor	Realizar el cálculo de las métricas por responsable	Alex Cunachi Erick Dávalos		

8. Sección 8: Modelo entidad-relación y modelo físico de la base de datos desarrollada del sistema antiguo

- Modelo entidad-relación de la base de datos desarrollada antes del mantenimiento de software



- Modelo físico de la base de datos desarrollada antes del mantenimiento de software



9. Sección 9: Historias de usuario

HU_01: Gestión del módulo de empresa productiva

Historia de Usuario	
Número: HU_01	Nombre de la Historia: Gestión del módulo de empresa productiva
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período Asignado: 2
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de una empresa productiva, para tener conocimiento de su nombre y a que se dedica	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice la empresa productiva por parroquias de Chimborazo • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de empresa productiva	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización de las empresas productivas de la provincia de Chimborazo
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 18/04/2022	Fecha Fin: 29/04/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo de empresas productivas de la provincia de Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de empresa productiva	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de empresas productivas de las Parroquias de la provincia de Chimborazo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 30/04/2022	Fecha Fin: 06/05/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de las empresas productivas de las parroquias de la provincia de Chimborazo.	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_01	Nombre de la Historia: Gestión del módulo de empresa productiva
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Alex Cunachi	Fecha: 09/05/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_01	Nombre de la Historia: Gestión del módulo de empresa productiva
Nombre: Verificar que se ingrese, modifique, elimine y visualice la empresa productiva por parroquias de Chimborazo	
Responsable: Alex Cunachi	Fecha: 09/05/2022
Descripción: Verificar que el módulo de empresas productivas se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de empresa productiva ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de la empresa productiva de la parroquia y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la empresa a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de las empresas productivas de la provincia de Chimborazo
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Alex Cunachi	Fecha: 16/05/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar los tipos de designación en las capas de interfaz de usuario y	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de las empresas productivas de la provincia de Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Alex Cunachi	Fecha: 16/05/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado: <ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de empresas productivas de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Alex Cunachi	Fecha: 16/05/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar los tipos de designación en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Dar clic en opción empresas productivas • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de empresas productivas de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Sandra Ordoñez	Fecha: 16/05/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de empresas productivas.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en tipo designación • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado:	
<ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

HU_02: Gestión del módulo de usuario

Historia de Usuario	
Número: HU_02	Nombre de la Historia: Gestión del módulo de usuario
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período asignado: 3
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de usuario, para tener conocimiento del tenido a ingresar	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice el usuario por parroquias de Chimborazo • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de usuario	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización de los usuarios de la provincia de Chimborazo
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 09/05/2022	Fecha Fin: 16/05/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo de usuarios de la provincia de Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de usuario	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación de usuarios de las Parroquias de la provincia de Chimborazo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 17/05/2022	Fecha Fin: 23/05/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de los usuarios de las parroquias de la provincia de Chimborazo.	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_02	Nombre de la Historia: Gestión del módulo de usuario
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Alex Cunachi	Fecha: 24/05/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_02	Nombre de la Historia: Gestión del módulo de usuario
Nombre: Verificar que se ingrese, modifique, elimine y visualice los usuarios de las parroquias de Chimborazo	
Responsable: Alex Cunachi	Fecha: 03/01/2019
Descripción: Verificar que el módulo de usuarios se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de usuarios ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Para modificar debe seleccionar el usuario a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de los usuarios de la provincia de Chimborazo
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Alex Cunachi	Fecha: 24/05/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar los tipos de designación en las capas de interfaz de usuario y	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de los usuarios de la provincia de Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Alex Cunachi	Fecha: 24/05/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado: <ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de los usuarios de los usuarios de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar	
Responsable: Alex Cunachi	Fecha: 27/05/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar los tipos de designación en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Ir al menú • Dar clic en opción de usuarios • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de empresas productivas de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Alex Cunachi	Fecha: 27/05/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de empresas productivas.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en usuarios • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

HU_03: Gestión del módulo de movilidad

Historia de Usuario	
Número: HU03	Nombre de la Historia: Gestión del módulo de movilidad
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período asignado: 3
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de movilidad de kilometraje de las vías que unen las parroquias de Chimborazo.	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice la información de movilidad por parroquias de Chimborazo • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de movilidad	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización del módulo de movilidad de la provincia de Chimborazo
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 23/05/2022	Fecha Fin: 30/05/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo de movilidad de la provincia de Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario Gestión del módulo de movilidad	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación de movilidad de las Parroquias de la provincia de Chimborazo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 31/05/2022	Fecha Fin: 03/06/2022
Programador Responsable: Erick Dávalos	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de la información de movilidad de las parroquias de la provincia de Chimborazo.	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_03	Nombre de la Historia: Gestión del módulo de movilidad
Nombre: Verificar que se ingrese, modifique, elimine y visualice la información de movilidad de las parroquias de Chimborazo	
Responsable: Alex Cunachi	Fecha: 06/06/2022
Descripción: Verificar que el módulo de movilidad se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	

<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de movilidad ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de movilidad de la parroquia y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la información de movilidad a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará
<p>Evaluación de la Prueba: Exitosa</p>

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_03	Nombre de la Historia: Gestión del módulo de movilidad
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Alex Cunachi	Fecha: 06/06/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de movilidad de la provincia de
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Alex Cunachi	Fecha: 06/06/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de movilidad de las parroquias de la provincia de Chimborazo en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de movilidad de la provincia de Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Alex Cunachi	Fecha: 06/06/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado: <ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de movilidad de los usuarios de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Alex Cunachi	Fecha: 09/06/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de movilidad en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Ir al menú • Dar clic en opción de movilidad • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de movilidad de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Alex Cunachi	Fecha: 09/06/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de movilidad.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en la opción de movilidad • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

HU_04: Gestión del módulo de económico social

Historia de Usuario	
Número: HU_04	Nombre de la Historia: Gestión del módulo de económico social
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período asignado: 4
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de económico social para tener registrado la infraestructura realizada con la ayuda de la institución pública	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice la información de económico social por parroquias de Chimborazo • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de económico social	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización módulo de económico social de la provincia de Chimborazo
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 06/06/2022	Fecha Fin: 13/06/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo de económico social de la provincia de Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario Gestión del módulo de económico social	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación de económico social de las Parroquias de la provincia de Chimborazo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 14/06/2022	Fecha Fin: 17/06/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de la información de económico social de las parroquias de la provincia de Chimborazo.	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_04	Nombre de la Historia: Gestión del módulo de económico social
Nombre: Verificar que se ingrese, modifique, elimine y visualice la información de económico social de las parroquias de Chimborazo	
Responsable: Erick Dávalos	Fecha: 20/06/2022
Descripción: Verificar que el módulo de económico social se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	

<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de económico social ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de movilidad de la parroquia y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la información de movilidad a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará
<p>Evaluación de la Prueba: Exitosa</p>

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_04	Nombre de la Historia: Gestión del módulo de económico social
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Erick Dávalos	Fecha: 20/06/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de económico social de la provincia de Chimborazo
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Erick Dávalos	Fecha: 20/06/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de económico social de las parroquias de la provincia de Chimborazo en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de económico social de la provincia de Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Erick Dávalos	Fecha: 20/06/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado: <ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de económico social de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar	
Responsable: Erick Dávalos	Fecha: 24/06/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de económico social en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Ir al menú • Dar clic en opción de económico social • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de económico social de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Erick Dávalos	Fecha: 24/06/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de económico social.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en la opción de económico social • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

HU_05: Gestión del módulo de generar solicitud

Historia de Usuario	
Número: HU_05	Nombre de la Historia: Gestión del módulo de generar solicitud
Modificación de historia de usuario:	
Usuario: Técnicos de las parroquias de Chimborazo	Período asignado 4: 1
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como técnicos de las parroquias de Chimborazo se desea realizar el registro de una solicitud a la institución de CONAGOPARE Chimborazo	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice la información de la solicitud y se pueda realizar la impresión de la misma en formato pdf. • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de generar solicitud	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización módulo de generar solicitud a la institución de CONAGOPARE
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 20/06/2022	Fecha Fin: 27/06/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo de generar de solicitud a CONAGOPARE Chimborazo	

PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario Gestión del módulo de generar solicitud	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación de generar solicitud a la institución de CONAGOPARE.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 28/06/2022	Fecha Fin: 01/07/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de la información de generar solicitud a la institución de CONAGOPARE	

PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

TAREA DE INGENIERÍA	
Historia de Usuario Gestión del módulo de generar solicitud	
Número de Tarea: TI_03	Nombre de Tarea: asignar un estado a una solicitud que se envía a la institución de CONAGOPARE Chimborazo
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 28/06/2022	Fecha Fin: 01/07/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito asignar un estado de espera de respuesta a la solicitud enviada a la institución de CONAGOPARE Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar la asignación del estado en el sistema es la correcta 	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_05	Nombre de la Historia: Gestión del módulo de generar solicitud
Nombre: Verificar que se ingrese, modifique, elimine y visualice la información de generar solicitud a la institución de CONAGOPARE	
Responsable: Erick Dávalos	Fecha: 05/07/2022
Descripción: Verificar que el módulo de generar solicitud se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	

<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de gestión documental ubicada en un menú el lado izquierdo del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de la solicitud emitida a la institución de CONAGOPARE Chimborazo y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la información de la solicitud a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará
<p>Evaluación de la Prueba: Exitosa</p>

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_05	Nombre de la Historia: Gestión del módulo de generar solicitud
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Erick Dávalos	Fecha: 05/07/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción realizada. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de generar solicitud a la institución de CONAGOPARE
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Erick Dávalos	Fecha: 19/12/2018
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de generar solicitud a la institución de CONAGOPARE Chimborazo en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de generar solicitud a la institución de CONAGOPARE Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Erick Dávalos	Fecha: 05/07/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado: <ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de generar solicitud a la institución de CONAGOPARE Chimborazo
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido.	
Responsable: Erick Dávalos	Fecha: 07/07/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información de generar solicitud a la institución de CONAGOPARE Chimborazo en las capas de interfaz de usuario y accesos a datos.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Ir al menú • Dar clic en opción de gestión documental • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de económico social de las Parroquias de la provincia de Chimborazo.
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad.	
Responsable: Erick Dávalos	Fecha: 07/07/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de generar solicitud a la institución de CONAGOPARE Chimborazo	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en la opción gestión documental • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_03	Tarea Ingeniería: Asignar un estado a una solicitud que se envía a la institución de CONAGOPARE Chimborazo
Nombre: • Verificar la asignación del estado en el sistema es la correcta	
Responsable: Erick Dávalos	Fecha: 09/07/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad para agregarle un estado de respuesta (en proceso, finalizado)al documento de solicitud emitida a la institución de CONAGOPARE Chimborazo	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema • Dar click en la opción gestión documental • Asignar un estado de respuesta a la solicitud • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La asignación de estado de respuesta se lo realizo con éxito 	
Evaluación de la Prueba: Exitosa	

HU_06: Gestión del módulo generar memorándums

Historia de Usuario	
Número: HU_06	Nombre de la Historia: Gestión del módulo de generar memorándums
Modificación de historia de usuario:	
Usuario: Presidente de CONAGOPARE Chimborazo	Período asignado: 5
Prioridad en el Negocio: Media	Puntos Estimados: 60
Riesgo en el Desarrollo: Medio	Puntos Reales: 60
Descripción: Como presidente de CONAGOPARE Chimborazo deseo emitir memorándums para asignación de proyectos a los integrantes de la institución	
Observaciones:	
Historia de Usuario (Reverso) Pruebas de Aceptación	
<ul style="list-style-type: none"> • Verificar que se ingrese, modifique, elimine y visualice la información de la solicitud y se pueda realizar la impresión de la misma en formato pdf. • Verificar que los datos ingresados, modificados o eliminados se guarden correctamente. 	

TAREA DE INGENIERÍA	
Historia de Usuario: Gestión del módulo de generar memorándums	
Número de Tarea: TI_01	Nombre de Tarea: Crear las clases para el ingreso, modificaciones, eliminación y visualización módulo generar memorándums
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 27/06/2022	Fecha Fin: 04/07/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear las clases necesarias para la gestión del módulo generar memorándums a los trabajadores de CONAGOPARE Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Comprobar que se hayan creado correctamente las clases. • Comprobar que cada clase tenga los atributos y métodos necesarios. 	

TAREA DE INGENIERÍA	
Historia de Usuario Gestión del módulo de generar memorándums	
Número de Tarea: TI_02	Nombre de Tarea: Crear la interfaz de usuarios para el ingreso, modificación y eliminación de los memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 30
Fecha Inicio: 05/07/2022	Fecha Fin: 08/07/2022
Programador Responsable: Alex Cunachi	
Descripción: Como desarrollador necesito crear la interfaz de usuario para el ingreso, modificación, y eliminación de la información de memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo	
PRUEBAS DE ACEPTACIÓN	
<ul style="list-style-type: none"> • Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido. • Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad. 	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-HU_06	Nombre de la Historia: Gestión del módulo de generar memorándums
Nombre: Verificar que se ingrese, modifique, elimine y visualice la información generar memorándums a los trabajadores de la institución de CONAGOPARE	
Responsable: Erick Dávalos	Fecha: 15/07/2022
Descripción: Verificar que el módulo de generar memorándums se pueda gestionar es decir guardar, eliminar, actualizar y visualizar cada uno de ellos.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Para realizar un ingreso debe dirigirse a la opción de administrador en el listado de las solicitudes ubicada en el listado central del sistema en el cual debe seleccionar una de las secciones presentadas. • Ingresar la información de los memorándums emitidas a los trabajadores de la institución de CONAGOPARE y darle clic en guardar automáticamente se agregará al listado ubicado en la parte central del sistema • Para modificar debe seleccionar la información de los memorándums a modificar del listado ubicado en el centro del sistema, los datos aparecerán en los campos pertinente a cada información, al terminar la modificación se debe dar clic en guardar. • Para poder eliminar del listado solo se deberá dar clic en el botón con el icono de eliminar y automáticamente la tabla se actualizará 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-HU_05	Nombre de la Historia: Gestión del módulo de generar memorándums
Nombre: Verificar que los datos ingresados, modificados y eliminados se guarden correctamente	
Responsable: Erick Dávalos	Fecha: 15/07/2022
Descripción: Comprobar que los datos ingresados, modificados y eliminados se guarden correctamente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Modificar, eliminar o ingresar los datos • Ingresar a la base de datos • Verificar que los datos se hayan alterado correctamente en base a la acción 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo
Nombre: Comprobar que se hayan creado correctamente las clases	
Responsable: Erick Dávalos	Fecha: 15/07/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, liminar y visualizar la información generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo en las capas de interfaz de usuario y accesos a datos	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Revisar que las clases existan en la capa de acceso • Debe contar con todas las clases y cada una de ellas se encuentra heredadas de las clases de clase común. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que las clases se hayan creado correctamente 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_02-TI_01	Tarea Ingeniería: Crear las clases para el ingreso, modificaciones, eliminación y visualización de generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo
Nombre: Comprobar que cada clase tenga los atributos y métodos necesarios.	
Responsable: Erick Dávalos	Fecha: 15/07/2022
Descripción: Se verificará que cada clase creada tenga los atributos y métodos necesarios.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que ya este creado el acceso a datos y la interfaz de usuario. • Que ya este creada la clase común. 	
Pasos de Ejecución:	
<ul style="list-style-type: none"> • Compara que todos los atributos estén acordes con la base de datos • Verificar que todos los métodos estén creados en base a la funcionalidad requerida y a los atributos antes definidos. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • Que se haya comprobado que cada clase creada cuente con los atributos y métodos necesarios. 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo de generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo
Nombre: Verificar si la interfaz de usuario correspondiente cumple con el estándar establecido	
Responsable: Erick Dávalos	Fecha: 19/07/2022
Descripción: Se verificará que las clases sean creadas correctamente para ingresar, modificar, eliminar y visualizar la información generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo en las capas de interfaz de usuario y accesos	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Controlador debe estar creado. • La vista debe estar creado. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Acceder al administrador dar clic en memorándums • Llenar los campos de ingreso • Verificar la interfaz 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

PRUEBA DE ACEPTACIÓN	
Código: PA_01-TI_02	Tarea Ingeniería: Crear la interfaz de usuarios para el ingreso, modificación y eliminación del módulo generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo
Nombre: Verificar que la interfaz tenga los campos necesarios requeridos por la funcionalidad	
Responsable: Erick Dávalos	Fecha: 19/07/2022
Descripción: Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad de ingreso, modificación, eliminación, y visualización de generar memorándums a los trabajadores de la institución de CONAGOPARE Chimborazo	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Verificar que la interfaz de usuario tenga los campos necesarios que requiere la funcionalidad. 	
Pasos de Ejecución: <ul style="list-style-type: none"> • Loguearse en el Sistema en modo administrador • Dar click en la opción de memorándums y generar uno nuevo • Compara los datos que requiere la funcionalidad con la interfaz de usuario 	
Resultado Esperado: <ul style="list-style-type: none"> • La interfaz cumple con el estándar establecido 	
Evaluación de la Prueba: Exitosa	

10. Sección 10: Resultados de las pruebas unitarias y de integración en el sistema de información productiva agropecuaria CONAGOPARE Chimborazo

All files

96.77% Statements 668/682 71.74% Branches 66/92 94.09% Functions 191/203 96.82% Lines 609/629

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
src	100%	11/11	100%	0/0
src/app	92.86%	13/14	50%	1/2
src/app/Servicios	100%	74/74	100%	0/0
src/auth/Admin-memo	93.33%	14/15	50%	1/2
src/auth/Admin-solicitud	97.78%	44/45	83.33%	5/6
src/auth/Admin-usuario	95.65%	22/23	66.67%	4/6
src/auth/Movilidad	96.77%	30/31	50%	1/2
src/auth/EcoSocial	97.09%	10/13	50%	1/2
src/auth/empresa	96.72%	29/31	50%	1/2
src/auth/solicitud	95.45%	21/24	72%	18/25
src/auth/parroquia	95.51%	25/29	72%	18/25
src/auth/turismo	96.91%	24/27	80%	16/20

11. Sección 11: Sistema de información productiva CONAGOPARE Chimborazo antes y después del mantenimiento de software

Interfaz de usuario del sistema de información productiva agropecuaria antes del mantenimiento de software

The screenshot displays the user interface of the 'SISTEMA DE CONTROL DE PRODUCCIÓN' (Production Control System). The interface is divided into a left sidebar with navigation options and a main content area.

Left Sidebar (Navigation):

- INFO PARROQUIA
- PROD AGRICOLA
- PROD PECUARIO
- INFRA GUBERNAMENTAL
- ESTACIÓN COMERCIO
- EMPRESAS PROD
- TURISMO
- CREDITOS

Main Content Area:

Listado de Productos Agrícolas

Buttons: Imprimir, Copiar, Excel. Search: Buscar:

Nombre	Tipo	Cantidad Hectáreas	Autoabastecimiento	Explotación	Comercio	Fecha		
MANZANA	FRUTA	44	66	75	549	2020-02-01		
MANZANA	FRUTA	34	53	72	47	2020-01-02		
PERA	FRUTA	56	76	87	55	2020-01-08		

Mostrando desde 1 al 3 de 3 elementos. Page navigation: Primero, Anterior, 1, Siguiente, Último.

Producto Agrícola Form:

Nombre:

Fecha:

Cantidad Hectáreas Producidas:

Autoabastecimiento:

Explotación:

Comercio:

Buttons: Limpiar Campos, AGREGAR, MODIFICAR

Interfaz de usuario del sistema de información productiva agropecuaria después del mantenimiento de software

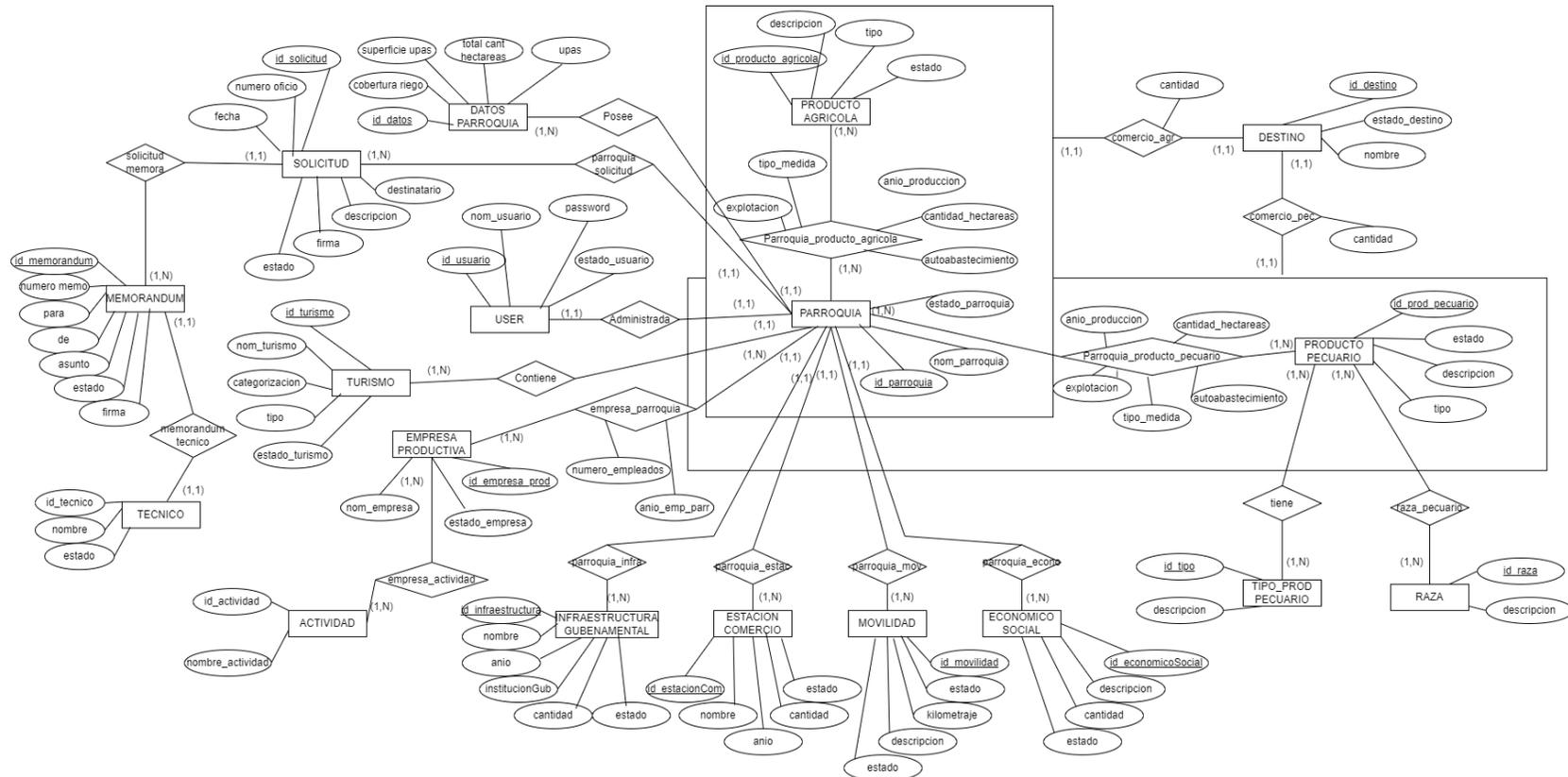
The screenshot displays the 'SISTEMA DE PRODUCCIÓN' interface. At the top, there is a blue header with a hamburger menu icon, the text 'SISTEMA DE PRODUCCIÓN', and a user profile icon. Below the header is a sidebar with various icons representing different system functions. The main content area is titled 'LISTADO DE PRODUCTOR AGRICOLAS' and includes three buttons: 'Imprimir', 'Copiar', and 'Excel'. A search bar labeled 'Buscar:' is positioned to the right. The central part of the interface is a table with the following columns: 'Nombre', 'Tipo', 'Cantidad Hectáreas', 'Autoabastecimiento', 'Explotación', 'Comercio', and 'Fecha'. Each row represents a different agricultural product, with icons for viewing and deleting each entry. Below the table, there is a pagination control showing 'Mostrando desde 1 al 5 de 5 elementos' and navigation buttons for 'Primero', 'Anterior', '1', 'Siguiete', and 'Último'. At the bottom, there is a section for 'PRODUCTO AGRICOLA' with a search field labeled 'NOMBRE'.

Nombre	Tipo	Cantidad Hectáreas	Autoabastecimiento	Explotación	Comercio	Fecha		
MANZANA	FRUTA	4	4	4	549	2022-06-11		
MANZANA	FRUTA	4	4	4	47	2022-06-11		
MANZANA	FRUTA	55	55	55	11	2022-06-11		
MANZANA	FRUTA	4	4	4	0	2022-06-11		
PERA	FRUTA	56	76	67	55	2020-01-08		

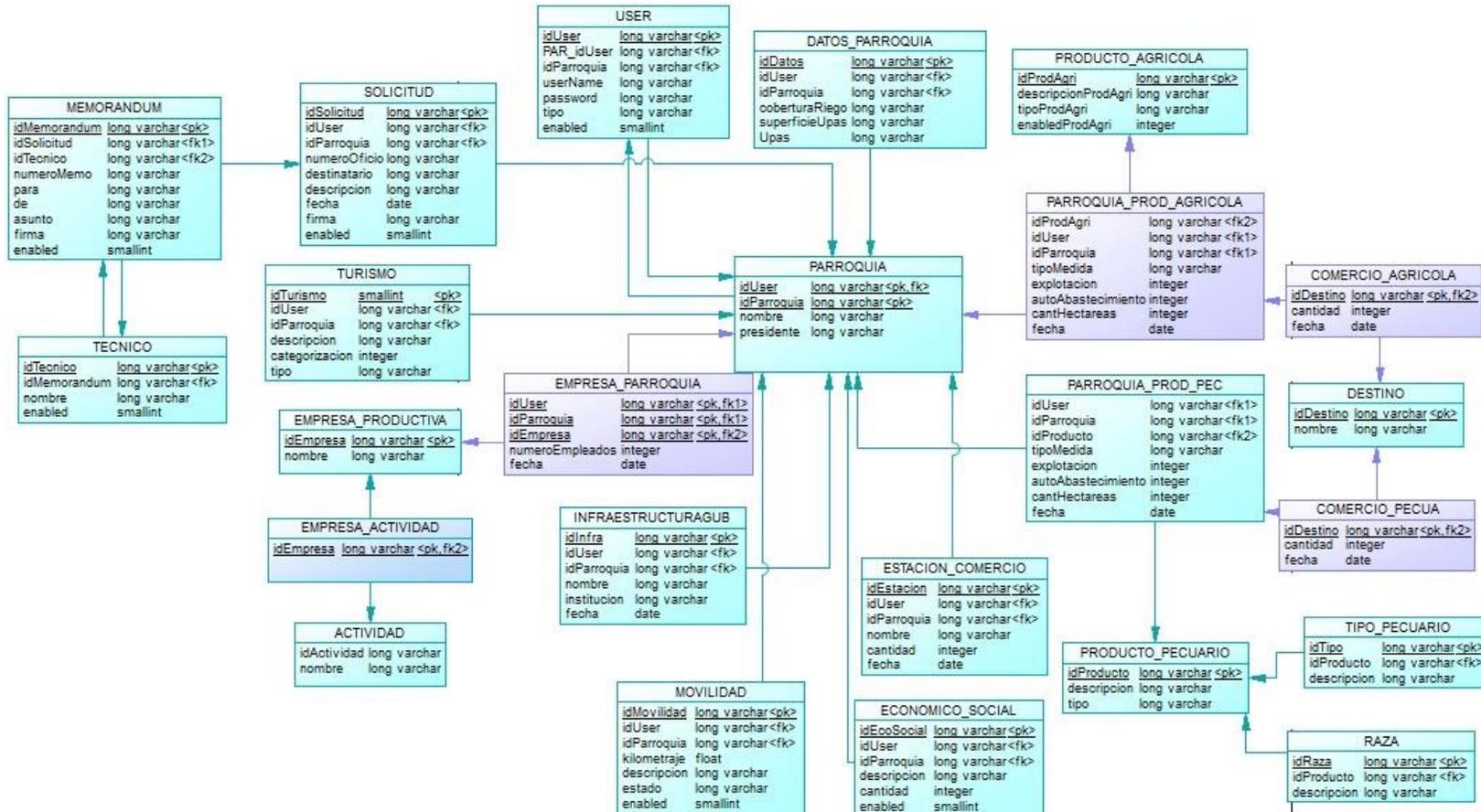
Se corrigieron errores en la interfaz de usuario se lo hizo más amigable al usuario y de manera Responsive con el motivo que se adapte a la pantalla del celular.

12. Sección 12: Modelo conceptual y físico de la base de datos actualizada después de aplicar mantenimiento de software

- Modelo entidad-relación de la base de datos actualizada después del mantenimiento de software



- Modelo físico de la base de datos actualizada después del mantenimiento de software



13. Sección 13: Diccionario de datos de la base de datos actualizada después del mantenimiento de software

ACTIVIDAD

Column	Type	Null	Default
ID_ACTIVIDAD (<i>Primary</i>)	int	No	
NOM_ACTIVIDAD	varchar(40)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_ACTIVIDAD	0	A	No

COMERCIO

Column	Type	Null	Default
id_comercio_agri (<i>Primary</i>)	bigint	No	
ID_DESTINO (<i>Primary</i>)	int	No	
ID_PRODUCTOS_AGRICOLA	int	No	
ID_PARROQUIA	int	No	
id_parr_prod_agri	int	No	
CANT_COMERCIO_AGRI	float	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_DESTINO	3	A	No
				id_comercio_agri	11	A	No
id_comercio_agri	BTREE	Yes	No	id_comercio_agri	11	A	No
FK_COMERCIO	BTREE	No	No	ID_PRODUCTOS_AGRICOLA	2	A	No
				ID_PARROQUIA	3	A	No
				ID_DESTINO	7	A	No
				id_parr_prod_agri	10	A	No

COMERCIO_PEC

Column	Type	Null	Default
id_comercio_pec (<i>Primary</i>)	bigint	No	
ID_DESTINO (<i>Primary</i>)	int	No	
ID_PARROQUIA	int	No	
ID_PRODUCTO_PECUARIA	int	No	
id_parr_prod_pec	int	No	
CANT_COMERCIO_PEC	float	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_DESTINO	3	A	No
				id_comercio_pec	18	A	No
id_comercio_pec	BTREE	Yes	No	id_comercio_pec	18	A	No
FK_COMERCIO_PEC2	BTREE	No	No	ID_PRODUCTO_PEC UARIA	4	A	No
				ID_PARROQUIA	4	A	No
				ID_DESTINO	8	A	No
				id_parr_prod_pec	16	A	No

DATOS_PARROQUIA

Column	Type	Null	Default
ID_DATOS_PARROQUIA (<i>Primary</i>)	int	No	
ID_PARROQUIA	int	No	
UPAS	float	No	
SUPERFICIE_UPAS	float	No	
COBERTURA_RIEGO	float	No	
TOTAL_CANT_HECTAREAS	float	No	
ANIO_DATOS_PARROQUIA	date	Yes	<i>NULL</i>

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_DATOS_PARR OQUIA	2	A	No
FK_DATOS_PARROQUIA	BTREE	No	No	ID_PARROQUIA	2	A	No

DESTINO

Column	Type	Null	Default
ID_DESTINO (<i>Primary</i>)	int	No	
NOM_DESTINO	varchar(40)	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_DESTINO	6	A	No

ECOSOCIAL

Column	Type	Null	Default
ID_ECOSOCIAL (<i>Primary</i>)	bigint	No	
ID_PARROQUIA	int	No	
DESCRIPCION	varchar(50)	No	
CANTIDAD	float	No	
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_ECOSOCIAL	6	A	No
FK_ECOSOCIAL_PARROQUIA	BTREE	No	No	ID_PARROQUIA	1	A	No

EMPRESA_ACTIVIDAD

Column	Type	Null	Default
ID_ACTIVIDAD (Primary)	int	No	
ID_EMP_PROD (Primary)	int	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_ACTIVIDAD	0	A	No
				ID_EMP_PROD	0	A	No
FK_EMPRESA_ACTIVIDAD2	BTREE	No	No	ID_EMP_PROD	0	A	No

EMPRESA_PARROQUIA

Column	Type	Null	Default
id_emp_parr (Primary)	bigint	No	
ID_PARROQUIA	int	No	
ID_EMP_PROD	int	No	
NUMERO_EMPLEADOS	int	No	
ANIO_EMPRESA_PARROQUIA	date	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_emp_parr	14	A	No
id_emp_parr	BTREE	Yes	No	id_emp_parr	14	A	No
FK_EMPRESA_PARROQUIA2	BTREE	No	No	ID_EMP_PROD	7	A	No
FK_EMPRESA_PARROQUIA	BTREE	No	No	ID_PARROQUIA	2	A	No

EMPRESA_PR

Column	Type	Null	Default
ID_EMP_PROD (Primary)	int	No	
NOM_EMPRESA	varchar(40)	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_EMP_PROD	10	A	No

ESTACION_COMERCIO

Column	Type	Null	Default
ID_ESTACION_COMERCIO (<i>Primary</i>)	int	No	
ID_PARROQUIA	int	No	
NOMBRE_ESTACION_COMERCIO	varchar(40)	No	
ANIO_ESTACION_COMERCIO	date	No	
CANT_ESTACION_COMERCIO	float	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_ESTACION_COMERCIO	14	A	No
FK_PARROQUIA_ESTACION_COMERCIO	BTREE	No	No	ID_PARROQUIA	1	A	No

INFRAESTRUCTURA_GUBERNAMENTAL

Column	Type	Null	Default
ID_INFRA_GUBER (<i>Primary</i>)	int	No	
ID_PARROQUIA	int	No	
NOM_INFRAESTRUCTURA	varchar(40)	No	
INSTITUCION_GUBERNAMENTAL	varchar(40)	No	
ANIO_INFRAESTRUCTURA	date	No	
CANT_INFRAESTRUCTURA	int	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation
PRIMARY	BTREE	Yes	No	ID_INFRA_GUBER	3	A
FK_PARROQUIA_INFRAESTRUCTURA_GUBERNAMENTAL	BTREE	No	No	ID_PARROQUIA	1	A

MEMORANDUM

Column	Type	Null	Default
ID_MEMORANDUM (<i>Primary</i>)	bigint	No	
ID_SOLICITUD	bigint	No	
NUMERO_MEMORANDUM	int	No	
FECHA_MEMORANDUM	date	Yes	<i>NULL</i>
PARA	varchar(50)	No	
DE	varchar(50)	No	
ASUNTO_MEMORANDUM	varchar(50)	No	
TECNICO	varchar(50)	No	
CONTENIDO_MEMORANDUM	varchar(999)	No	
FIRMA_MEMORANDUM	varchar(50)	No	
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_MEMORANDUM	0	A	No
ID_SOLICITUD	BTREE	No	No	ID_SOLICITUD	0	A	No

MOVILIDAD

Column	Type	Null	Default
ID_MOVILIDAD (<i>Primary</i>)	bigint	No	
ID_PARROQUIA	int	No	
DESCRIPCION	varchar(50)	No	
KILOMETRAJE	float	No	
ENABLED_MOVILIDAD	tinyint(1)	No	1
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_MOVILIDAD	10	A	No
FK_MOVILIDAD_PARROQUIA	BTREE	No	No	ID_PARROQUIA	1	A	No

PARROQUIA

Column	Type	Null	Default
ID_PARROQUIA (<i>Primary</i>)	int	No	
NOM_PARROQUIA	varchar(40)	No	
ENABLED	tinyint(1)	No	
PRESI_PARROQUIA	varchar(40)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_PARROQUIA	46	A	No

PARROQUIA_PRODUCTO_PECUARIO

Column	Type	Null	Default
id_parr_prod_pec (<i>Primary</i>)	bigint	No	
ID_PARROQUIA	int	No	
ID_PRODUCTO_PECUARIA	int	No	
ANIO_PARROQUIA_PRODUCTO_PECUARIO	date	Yes	NULL
CANT_PARROQUIA_PRODUCTO_PECUARIO	float	No	0
AUTOABASTECIMIENTO_DISTRIBUCION_PEC	float	No	
EXPLOTACION_DISTRIBUCION_PEC	float	No	
TIPO_MEDIADA_PEC	int	No	
ENABLED	tinyint(1)	No	1
RAZA	varchar(30)	Yes	NULL

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_parr_prod_pec	11	A	No
id_parr_prod_pec	BTREE	Yes	No	id_parr_prod_pec	11	A	No
FK_OBTENER	BTREE	No	No	ID_PARROQUIA	1	A	No
FK_DISTRIBUCION_PEC	BTREE	No	No	ID_PRODUCTO_P ECUARIA	4	A	No

PARROQUIA_PRUDUCTO_AGRICOLA

Column	Type	Null	Default
id_parr_prod_agri (<i>Primary</i>)	bigint	No	
ID_PRODUCTOS_AGRICOLA	int	No	
ID_PARROQUIA	int	No	
ANIO_PARROQUIA_PRODUCTO_AGRICOLA	date	No	
CANT_HECTAREAS_PRODUCIDAS	float	No	
AUTOABASTECIMIENTO_DISTRIBUCION_AGR	float	No	
EXPLOTACION_DISTRIBUCION_AGRI	float	No	
TIPO_MEDIDA	int	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	id_parr_prod_agri	9	A	No
id_parr_prod_agri	BTREE	Yes	No	id_parr_prod_agri	9	A	No
FK_TIENE	BTREE	No	No	ID_PARROQUIA	2	A	No
ID_PRODUCTOS_AGRICOLA	BTREE	No	No	ID_PRODUCTOS _AGRICOLA	3	A	No

PRODUCTO_AGRICOLA

Column	Type	Null	Default
ID_PRODUCTOS_AGRICOLA (<i>Primary</i>)	int	No	
DESCRIPCION	varchar(40)	No	
TIPO_PRODUCTOS_AGRICOLA	varchar(40)	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_PRODUCTOS_AGRICOLA	6	A	No

PRODUCTO_PECUARIA

Column	Type	Null	Default
ID_PRODUCTO_PECUARIA (<i>Primary</i>)	int	No	
DESCRIPCION	varchar(40)	No	
tipo_prod_pec	varchar(50)	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_PRODUCTO_PECUARIA	5	A	No

RAZA

Column	Type	Null	Default
ID_RAZA (<i>Primary</i>)	int	No	
ID_PRODUCTO_PECUARIA	int	No	
DESCRIPCION	varchar(40)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_RAZA	0	A	No
FK_PRODUCTO_PECUARIA_RAZA	BTREE	No	No	ID_PRODUCTO_PECUARIA	0	A	No

SOLICITUD

Column	Type	Null	Default
ID_SOLICITUD (<i>Primary</i>)	bigint	No	
ID_PARROQUIA	int	No	
NUMERO_OFICIO	int	No	
FECHA_SOLICITUD	date	Yes	<i>NULL</i>
DESTINATARIO	varchar(50)	No	
CONTENIDO	varchar(999)	No	
FIRMA	varchar(50)	No	
EDITABLE	tinyint(1)	No	1
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_SOLICITUD	0	A	No
FK_SOLICITUD_PARROQUIA	BTREE	No	No	ID_PARROQUIA	0	A	No

TECNICO

Column	Type	Null	Default
ID_TECNICO (<i>Primary</i>)	bigint	No	
NOMBRE	varchar(30)	No	
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_TECNICO	3	A	No

TIPO_PECUARIA

Column	Type	Null	Default
ID_TIPO_PECUARIO (<i>Primary</i>)	int	No	
DESCRIPCION	varchar(40)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_TIPO_PECUARIO	0	A	No

TIPO_PRODUCTO_PECUARIO

Column	Type	Null	Default
ID_PRODUCTO_PECUARIA (<i>Primary</i>)	int	No	
ID_TIPO_PECUARIO (<i>Primary</i>)	int	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_PRODUCTO_PECUARIA	0	A	No
				ID_TIPO_PECUARIO	0	A	No
FK_TIPO_PRODUCTO_PECUARIO2	BTREE	No	No	ID_TIPO_PECUARIO	0	A	No

TURISMO

Column	Type	Null	Default
ID_TURISMO (<i>Primary</i>)	int	No	
ID_PARROQUIA	int	No	
NOM_TURISMO	varchar(40)	No	
TIPO_TURISMO	varchar(40)	No	
CATEGORIZACION_TURISMO	int	No	
ENABLED	tinyint(1)	No	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_TURISMO	4	A	No
FK_TURISMO_PARROQUIA	BTREE	No	No	ID_PARROQUIA	1	A	No

USER

Column	Type	Null	Default
ID_USUARIO (<i>Primary</i>)	bigint	No	
username	varchar(30)	No	
pwuser	varchar(200)	No	
ID_PARROQUIA	int	No	
ENABLED	tinyint(1)	No	1

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
PRIMARY	BTREE	Yes	No	ID_USUARIO	46	A	No
FK_USER_PARROQUIA2	BTREE	No	No	ID_PARROQUIA	46	A	No

Anexo B: Valores obtenidos de las pruebas de penetración a las vulnerabilidades del software



VALORES OBTENIDOS DE LAS PRUEBAS DE PENETRACIÓN A LAS
VULNERABILIDADES DEL SOFTWARE

AUTORES:

ERICK FABRICIO DÁVALOS MANCERO

ALEX SANTIAGO CUNACHI CHACHA

Riobamba – Ecuador

2022

Tabla de contenidos

Introducción	1
1. Pruebas realizadas antes del mantenimiento software	2
1.1. Pruebas de Directorio transversal	2
1.2. Pruebas de Cross-site request forgery (CSRF)	5
1.3. Ataque de intercepción criptográfica	9
1.4. Pruebas de SQL Injection	9
1.5. Pruebas Cross-site scripting (XSS)	13
1.6. Ataque de password mismanagement	16
1.7. Ataque de privilege escalation	17
2. Pruebas realizadas después del mantenimiento de software	22
2.1. Pruebas de Directorio transversal	22
2.2. Pruebas de Cross-site request forgery (CSRF)	25
2.3. Ataque de intercepción criptográfica	28
2.4. Pruebas de SQL Injection	29
2.5. Pruebas Cross-site scripting (XSS)	32
2.6. Ataque de password mismanagement	36
2.7. Ataque de privilege escalation	36

Introducción

Las pruebas de penetración (también llamadas “pen testing”) son una práctica para poner a prueba un sistema informático, red o aplicación web para encontrar vulnerabilidades que un atacante podría explotar.

El principal objetivo de las pruebas de penetración consiste en determinar las debilidades de seguridad. Una prueba de penetración también puede ser utilizada para probar el cumplimiento de la política de seguridad de una organización, la conciencia de seguridad de sus empleados y la capacidad de la organización para identificar y responder a los incidentes de seguridad.

En este caso la prueba de penetración se realizó para probar la seguridad del sistema de información productiva agropecuaria CONAGOPARE Chimborazo con la finalidad de obtener resultados ante el software desarrollado antiguamente y el que se obtiene después de haber aplicado el mantenimiento de software y así poder verificar si al aplicar las buenas prácticas de seguridad el sistema tuvo mejoras en cuanto a la seguridad

1. Pruebas realizadas antes del mantenimiento software

1.1. Pruebas de Directorio transversal

Nombre	Ingresar información de la parroquia (Directorio transversal)	Identificador		T1
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar información de la parroquia (Directorio transversal)	Identificador		T2
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto agrícola (Directorio transversal)	Identificador		T3
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto agrícola (Directorio transversal)	Identificador		T4
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto pecuario (Directorio transversal)	Identificador		T5
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto pecuario (Directorio transversal)	Identificador		T6
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar infraestructura gubernamental (Directorio transversal)	Identificador		T7
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (Directorio transversal)	Identificador		T8
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar estación de comercio (Directorio transversal)	Identificador		T9
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar estación de comercio (Directorio transversal)	Identificador		T10
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar sitio turístico (Directorio transversal)	Identificador		T11
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (Directorio transversal)	Identificador		T12
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T1	Ingresar información de la parroquia	0% Vulnerable
T2	Modificar información de la parroquia	0% Vulnerable
T3	Ingresar producto agrícola	0% Vulnerable
T4	Modificar producto agrícola	0% Vulnerable
T5	Ingresar producto pecuario	0% Vulnerable
T6	Modificar producto pecuario	0% Vulnerable
T7	Ingresar infraestructura gubernamental	0% Vulnerable
T8	Obtener infraestructura gubernamental	0% Vulnerable
T9	Ingresar estación de comercio	0% Vulnerable
T10	Modificar estación de comercio	0% Vulnerable
T11	Ingresar sitio turístico	0% Vulnerable
T12	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

1.2. Pruebas de Cross-site request forgery (CSRF)

Nombre	Ingresar información de la parroquia (CSRF)	Identificador	T13
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Modificar información de la parroquia (CSRF)	Identificador	T14
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Ingresar producto agrícola (CSRF)	Identificador	T15
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Modificar producto agrícola (CSRF)	Identificador	T16
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Ingresar producto pecuario (CSRF)	Identificador	T17
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Modificar producto pecuario (CSRF)	Identificador	T18
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Ingresar infraestructura gubernamental (CSRF)	Identificador		T19
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener infraestructura gubernamental (CSRF)	Identificador		T20
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Ingresar estación de comercio (CSRF)	Identificador		T21
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Modificar estación de comercio (CSRF)	Identificador		T22
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Ingresar sitio turístico (CSRF)	Identificador	T23
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Modificar sitio turístico (CSRF)	Identificador	T24
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T13	Ingresar información de la parroquia	100% Vulnerable
T14	Modificar información de la parroquia	100% Vulnerable
T15	Ingresar producto agrícola	100% Vulnerable
T16	Modificar producto agrícola	100% Vulnerable
T17	Ingresar producto pecuario	100% Vulnerable
T18	Modificar producto pecuario	100% Vulnerable
T19	Ingresar infraestructura gubernamental	100% Vulnerable
T20	Obtener infraestructura gubernamental	100% Vulnerable
T21	Ingresar estación de comercio	100% Vulnerable
T22	Modificar estación de comercio	100% Vulnerable
T23	Ingresar sitio turístico	100% Vulnerable
T24	Modificar sitio turístico	100% Vulnerable
Promedio total		100%

1.3. Ataque de intercepción criptográfica

Nombre	Autenticación (Intercepción criptográfica)		Identificador	T25
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T25	Autenticación	100% Vulnerable
Promedio total		100%

1.4. Pruebas de SQL Injection

Nombre	Seleccionar información de la parroquia (SQL Injection)		Identificador	T26
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener información de las parroquias (SQL Injection)		Identificador	T27
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar producto agrícola (SQL Injection)	Identificador		T28
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener productos agrícolas (SQL Injection)	Identificador		T29
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar producto pecuario (SQL Injection)	Identificador		T30
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener productos pecuarios (SQL Injection)	Identificador		T31
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar infraestructura gubernamental (SQL Injection)		Identificador	T32
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener infraestructura gubernamental (SQL Injection)		Identificador	T33
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar estación de comercio (SQL Injection)		Identificador	T34
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener estaciones de comercio (SQL Injection)		Identificador	T35
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar sitio turístico (SQL Injection)		Identificador	T36
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener sitios turísticos (SQL Injection)		Identificador	T37
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T26	Seleccionar información de la parroquia	100% Vulnerable
T27	Obtener información de las parroquias	100% Vulnerable
T28	Seleccionar producto agrícola	100% Vulnerable
T29	Obtener productos agrícolas	100% Vulnerable
T30	Seleccionar producto pecuario	100% Vulnerable
T31	Obtener productos pecuarios	100% Vulnerable
T32	Seleccionar infraestructura gubernamental	100% Vulnerable
T33	Obtener infraestructura gubernamental	100% Vulnerable
T34	Seleccionar estación de comercio	100% Vulnerable
T35	Obtener estaciones de comercio	100% Vulnerable
T36	Seleccionar sitio turístico	100% Vulnerable
T37	Obtener sitios turísticos	100% Vulnerable
Promedio total		100%

1.5. Pruebas Cross-site scripting (XSS)

Nombre	Ingresar información de la parroquia (XSS)	Identificador		T38
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar información de la parroquia (XSS)	Identificador		T39
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto agrícola (XSS)	Identificador		T40
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto agrícola (XSS)	Identificador		T41
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto pecuario (XSS)	Identificador		T42
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto pecuario (XSS)	Identificador		T43
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar infraestructura gubernamental (XSS)	Identificador		T44
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (XSS)	Identificador		T45
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar estación de comercio (XSS)	Identificador		T46
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar estación de comercio (XSS)	Identificador		T47
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar sitio turístico (XSS)	Identificador		T48
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (XSS)	Identificador		T49
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T38	Ingresar información de la parroquia	0% Vulnerable
T39	Modificar información de la parroquia	0% Vulnerable
T40	Ingresar producto agrícola	0% Vulnerable
T41	Modificar producto agrícola	0% Vulnerable
T42	Ingresar producto pecuario	0% Vulnerable
T43	Modificar producto pecuario	0% Vulnerable
T44	Ingresar infraestructura gubernamental	0% Vulnerable
T45	Obtener infraestructura gubernamental	0% Vulnerable
T46	Ingresar estación de comercio	0% Vulnerable
T47	Modificar estación de comercio	0% Vulnerable
T48	Ingresar sitio turístico	0% Vulnerable
T49	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

1.6. Ataque de password mismanagement

Nombre	Autenticación (password mismanagement)		Identificador	T50
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T50	Autenticación	100% Vulnerable
Promedio total		100%

1.7. Ataque de privilege escalation

Nombre	Modificar información de la parroquia (privilege escalation)	Identificador		T51
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Eliminar información de la parroquia (privilege escalation)	Identificador		T52
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar información de la parroquia (privilege escalation)	Identificador		T53
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener información de las parroquias (privilege escalation)	Identificador		T54
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Eliminar producto agrícola (privilege escalation)		Identificador	T55
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Seleccionar producto agrícola (privilege escalation)		Identificador	T56
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Ingresar producto pecuario (privilege escalation)		Identificador	T57
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener productos pecuarios (privilege escalation)		Identificador	T58
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Ingresar infraestructura gubernamental (privilege escalation)	Identificador	T59
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Eliminar infraestructura gubernamental (privilege escalation)	Identificador	T60
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Seleccionar infraestructura gubernamental (privilege escalation)	Identificador	T61
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Obtener infraestructura gubernamental (privilege escalation)	Identificador	T62
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 100% vulnerable		
Estado	Vulnerable:	X	No Vulnerable:

Nombre	Seleccionar estación de comercio (privilege escalation)	Identificador		T63
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Obtener estaciones de comercio (privilege escalation)	Identificador		T64
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Modificar sitio turístico (privilege escalation)	Identificador		T65
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

Nombre	Eliminar sitio turístico (privilege escalation)	Identificador		T66
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 100% vulnerable			
Estado	Vulnerable:	X	No Vulnerable:	

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T51	Modificar información de la parroquia	100% Vulnerable
T52	Eliminar información de la parroquia	100% Vulnerable
T53	Seleccionar información de la parroquia	100% Vulnerable
T54	Obtener información de las parroquias	100% Vulnerable
T55	Eliminar producto agrícola	100% Vulnerable
T56	Seleccionar producto agrícola	100% Vulnerable
T57	Ingresar producto pecuario	100% Vulnerable
T58	Obtener productos pecuarios	100% Vulnerable
T59	Ingresar infraestructura gubernamental	100% Vulnerable
T60	Eliminar infraestructura gubernamental	100% Vulnerable
T61	Seleccionar infraestructura gubernamental	100% Vulnerable
T62	Obtener infraestructura gubernamental	100% Vulnerable
T63	Seleccionar estación de comercio	100% Vulnerable
T64	Obtener estaciones de comercio	100% Vulnerable
T65	Modificar sitio turístico	100% Vulnerable
T66	Eliminar sitio turístico	100% Vulnerable
Promedio total		100%

2. Pruebas realizadas después del mantenimiento de software

2.1. Pruebas de Directorio transversal

Nombre	Ingresar información de la parroquia (Directorio transversal)		Identificador	T1
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar información de la parroquia (Directorio transversal)		Identificador	T2
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto agrícola (Directorio transversal)		Identificador	T3
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto agrícola (Directorio transversal)		Identificador	T4
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto pecuario (Directorio transversal)	Identificador		T5
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto pecuario (Directorio transversal)	Identificador		T6
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar infraestructura gubernamental (Directorio transversal)	Identificador		T7
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (Directorio transversal)	Identificador		T8
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar estación de comercio (Directorio transversal)	Identificador		T9
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar estación de comercio (Directorio transversal)	Identificador		T10
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar sitio turístico (Directorio transversal)	Identificador		T11
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (Directorio transversal)	Identificador		T12
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T1	Ingresar información de la parroquia	0% Vulnerable
T2	Modificar información de la parroquia	0% Vulnerable
T3	Ingresar producto agrícola	0% Vulnerable
T4	Modificar producto agrícola	0% Vulnerable
T5	Ingresar producto pecuario	0% Vulnerable
T6	Modificar producto pecuario	0% Vulnerable
T7	Ingresar infraestructura gubernamental	0% Vulnerable
T8	Obtener infraestructura gubernamental	0% Vulnerable
T9	Ingresar estación de comercio	0% Vulnerable
T10	Modificar estación de comercio	0% Vulnerable
T11	Ingresar sitio turístico	0% Vulnerable
T12	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

2.2. Pruebas de Cross-site request forgery (CSRF)

Nombre	Ingresar información de la parroquia (CSRF)	Identificador	T13
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Modificar información de la parroquia (CSRF)	Identificador	T14
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Ingresar producto agrícola(CSRF)	Identificador	T15
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:		No Vulnerable: X

Nombre	Modificar producto agrícola (CSRF)	Identificador	T16
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:		No Vulnerable: X

Nombre	Ingresar producto pecuario (CSRF)	Identificador	T17
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:		No Vulnerable: X

Nombre	Modificar producto pecuario (CSRF)	Identificador	T18
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:		No Vulnerable: X

Nombre	Ingresar infraestructura gubernamental (CSRF)	Identificador	T19
Valor máximo	100%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:		No Vulnerable: X

Nombre	Obtener infraestructura gubernamental (CSRF)	Identificador		T20
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar estación de comercio (CSRF)	Identificador		T21
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar estación de comercio (CSRF)	Identificador		T22
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar sitio turístico (CSRF)	Identificador		T23
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (CSRF)	Identificador		T24
Valor máximo	100%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T13	Ingresar información de la parroquia	0% Vulnerable
T14	Modificar información de la parroquia	0% Vulnerable
T15	Ingresar producto agrícola	0% Vulnerable
T16	Modificar producto agrícola	0% Vulnerable
T17	Ingresar producto pecuario	0% Vulnerable
T18	Modificar producto pecuario	0% Vulnerable
T19	Ingresar infraestructura gubernamental	0% Vulnerable
T20	Obtener infraestructura gubernamental	0% Vulnerable
T21	Ingresar estación de comercio	0% Vulnerable
T22	Modificar estación de comercio	0% Vulnerable
T23	Ingresar sitio turístico	0% Vulnerable
T24	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

2.3. Ataque de interceptación criptográfica

Nombre	Autenticación (Intercepción criptográfica)	Identificador		T25
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T25	Autenticación	0% Vulnerable
Promedio total		0%

2.4. Pruebas de SQL Injection

Nombre	Seleccionar información de la parroquia (SQL Injection)	Identificador	T26
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Obtener información de las parroquias (SQL Injection)	Identificador	T27
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Seleccionar producto agrícola (SQL Injection)	Identificador	T28
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Obtener productos agrícolas (SQL Injection)	Identificador	T29
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Seleccionar producto pecuario (SQL Injection)	Identificador		T30
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener productos pecuarios (SQL Injection)	Identificador		T31
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar infraestructura gubernamental (SQL Injection)	Identificador		T32
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (SQL Injection)	Identificador		T33
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar estación de comercio (SQL Injection)	Identificador		T34
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener estaciones de comercio (SQL Injection)	Identificador		T35
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar sitio turístico (SQL Injection)	Identificador		T36
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener sitios turísticos (SQL Injection)	Identificador		T37
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T26	Seleccionar información de la parroquia	0% Vulnerable
T27	Obtener información de las parroquias	0% Vulnerable
T28	Seleccionar producto agrícola	0% Vulnerable
T29	Obtener productos agrícolas	0% Vulnerable
T30	Seleccionar producto pecuario	0% Vulnerable
T31	Obtener productos pecuarios	0% Vulnerable
T32	Seleccionar infraestructura gubernamental	0% Vulnerable
T33	Obtener infraestructura gubernamental	0% Vulnerable
T34	Seleccionar estación de comercio	0% Vulnerable
T35	Obtener estaciones de comercio	0% Vulnerable
T36	Seleccionar sitio turístico	0% Vulnerable
T37	Obtener sitios turísticos	0% Vulnerable
Promedio total		0%

2.5. Pruebas Cross-site scripting (XSS)

Nombre	Ingresar información de la parroquia (XSS)	Identificador	T38
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Modificar información de la parroquia (XSS)	Identificador	T39
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Ingresar producto agrícola (XSS)	Identificador		T40
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto agrícola (XSS)	Identificador		T41
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar producto pecuario (XSS)	Identificador		T42
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar producto pecuario (XSS)	Identificador		T43
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar infraestructura gubernamental (XSS)	Identificador		T44
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (XSS)	Identificador		T45
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar estación de comercio (XSS)	Identificador		T46
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar estación de comercio (XSS)	Identificador		T47
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar sitio turístico (XSS)	Identificador		T48
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (XSS)	Identificador		T49
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T38	Ingresar información de la parroquia	0% Vulnerable
T39	Modificar información de la parroquia	0% Vulnerable
T40	Ingresar producto agrícola	0% Vulnerable
T41	Modificar producto agrícola	0% Vulnerable
T42	Ingresar producto pecuario	0% Vulnerable
T43	Modificar producto pecuario	0% Vulnerable
T44	Ingresar infraestructura gubernamental	0% Vulnerable
T45	Obtener infraestructura gubernamental	0% Vulnerable
T46	Ingresar estación de comercio	0% Vulnerable
T47	Modificar estación de comercio	0% Vulnerable
T48	Ingresar sitio turístico	0% Vulnerable
T49	Modificar sitio turístico	0% Vulnerable
Promedio total		0%

2.6. Ataque de password mismanagement

Nombre	Autenticación (password mismanagement)		Identificador	T50
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T50	Autenticación	0% Vulnerable
Promedio total		0%

2.7. Ataque de privilege escalation

Nombre	Modificar información de la parroquia (privilege escalation)		Identificador	T51
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Eliminar información de la parroquia (privilege escalation)		Identificador	T52
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar información de la parroquia (privilege escalation)	Identificador	T53
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Obtener información de las parroquias (privilege escalation)	Identificador	T54
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Eliminar producto agrícola (privilege escalation)	Identificador	T55
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Seleccionar producto agrícola (privilege escalation)	Identificador	T56
Valor máximo	0%	Valor mínimo	0%
Resultado esperado	Funcionalidad 0% vulnerable		
Resultados obtenidos	Funcionalidad 0% vulnerable		
Estado	Vulnerable:	No Vulnerable:	X

Nombre	Ingresar producto pecuario (privilege escalation)	Identificador		T57
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener productos pecuarios (privilege escalation)	Identificador		T58
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Ingresar infraestructura gubernamental (privilege escalation)	Identificador		T59
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Eliminar infraestructura gubernamental (privilege escalation)	Identificador		T60
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar infraestructura gubernamental (privilege escalation)		Identificador	T61
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener infraestructura gubernamental (privilege escalation)		Identificador	T62
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Seleccionar estación de comercio (privilege escalation)		Identificador	T63
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Obtener estaciones de comercio (privilege escalation)		Identificador	T64
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Modificar sitio turístico (privilege escalation)		Identificador	T65
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

Nombre	Eliminar sitio turístico (privilege escalation)		Identificador	T66
Valor máximo	0%	Valor mínimo	0%	
Resultado esperado	Funcionalidad 0% vulnerable			
Resultados obtenidos	Funcionalidad 0% vulnerable			
Estado	Vulnerable:		No Vulnerable:	X

ID	CASO DE USO	RESULTADO DE LA PRUEBA
T51	Modificar información de la parroquia	0% Vulnerable
T52	Eliminar información de la parroquia	0% Vulnerable
T53	Seleccionar información de la parroquia	0% Vulnerable
T54	Obtener información de las parroquias	0% Vulnerable
T55	Eliminar producto agrícola	0% Vulnerable
T56	Seleccionar producto agrícola	0% Vulnerable
T57	Ingresar producto pecuario	0% Vulnerable
T58	Obtener productos pecuarios	0% Vulnerable
T59	Ingresar infraestructura gubernamental	0% Vulnerable
T60	Eliminar infraestructura gubernamental	0% Vulnerable
T61	Seleccionar infraestructura gubernamental	0% Vulnerable
T62	Obtener infraestructura gubernamental	0% Vulnerable
T63	Seleccionar estación de comercio	0% Vulnerable
T64	Obtener estaciones de comercio	0% Vulnerable
T65	Modificar sitio turístico	0% Vulnerable
T66	Eliminar sitio turístico	0% Vulnerable
Promedio total		0%

Anexo C: Manual de buenas prácticas de seguridad



MANUAL DE BUENAS PRÁCTICAS

APLICACIONES WEB DESARROLLADAS EN ANGULAR CON PHP

AUTORES:

ERICK FABRICIO DÁVALOS MANCERO

ALEX SANTIAGO CUNACHI CHACHA

Riobamba – Ecuador

2022

Tabla de contenidos

Introducción	1
Contenido	1
Requerimientos software	1
1. Buenas prácticas de seguridad	2
1.1. SQL Injection	2
1.2. Cross site scripting (XSS)	3
1.3. Directorio transversal	4
1.4. Intercepción criptográfica	5
1.5. Cross-Site Request forgery	7
1.6. Password Mismanagement	8
1.7. Privilege Escalation	9
1.8. Session Fixation	10

Introducción

El propósito de este manual de buenas prácticas es facilitar al desarrollador técnicas que permiten mitigar 7 vulnerabilidades que los sistemas informáticos poseen, la aplicación de las técnicas se realizó en el sistema de producción agropecuaria.

Contenido

- Implementación del sistema
- Requerimientos hardware

Contar con:

- Computador personal.
- Conexión a Internet.

Requerimientos software

Contar con:

- Sistema operativo Windows.
- Navegador (Internet Explorer, Google Chrome, Mozilla Firefox, etc.).
- NodeJS.
- NPM : Node Package Manager.
- Typescript
- AngularCli
- Editor de texto(bloc de notas, visual studio code, notepad++,etc.)

1. Buenas prácticas de seguridad

1.1. SQL Injection

Es un ataque en contra de bases de datos como se observa en la Figura 4-1, el mismo que mediante parámetros de entrada o URL's agrega código SQL (lenguaje de consulta estructurado) con el objetivo de acceder o manipular la base de datos (OWASP Foundation, 2017).

1.1.1. Técnica para prevenir SQL Injection

Usar PreparedStatements en lugar de Statements en la recuperación de consultas SQL.

```
$stmt = $conexion->prepare('SELECT * FROM `datos_parroquia` WHERE ID_PARROQUIA = ?');  
$stmt->bind_param('s', $_GET['gidParroquia']);  
$stmt->execute();
```

Ilustración 1: PreparedStatement en Php

Validar entradas en el lado del cliente mediante una función llamada sanitize

```
sanitize(string) {  
    const map = {  
        '&': '&amp;',  
        '<': '&lt;',  
        '>': '&gt;',  
        '"': '&quot;',  
        "'": '&#x27;',  
        '/': '&#x2F;',  
    };  
    const reg = /[&<>"'/]/ig;  
    return string.replace(reg, (match)=>(map[match]));  
}
```

Ilustración 2: Función sanitize

```

comprobadorInput(forma: NgForm){
  let resultado = false;
  for (const field in forma.controls) {
    if(forma.controls[field].value !== this.sanitize(forma.controls[field].value)){
      resultado = true;
      return resultado;
    }
  }
  return resultado;
}

alta(forma: NgForm) {
  if (this.comprobadorInput(forma)) {
    Swal.fire('Error !', 'No insertes caracteres sospechosos', 'error');
    return;
  }
}

```

Ilustración 3: Uso de la función sanitize

1.2. Cross site scripting (XSS)

Este ataque fue creado para vulnerar aplicaciones web, la misma que permite vulnerar el sistema mediante inserción de código JavaScript y como consecuencia el atacante podrá robar contraseñas, redirigir a otros sitios web falsos, entre otros (Soto, 2016).

1.2.1. Técnica para prevenir Cross site scripting (XSS)

Validación de entradas con la filtración de caracteres especiales en parámetros de formato establecido y filtración de etiquetas HTML, con un complemento que nos permite controlar las entradas de texto

Paso 1: instalar el plugin inputmask

npm install --save inputmask

Paso 2: crear una directiva para envolver la máscara de entrada

```

import {Directive, ElementRef, Input} from '@angular/core';
import * as Inputmask from 'inputmask';

@Directive({
  selector: '[app-restrict-input]',
})
export class RestrictInputDirective {
  private regexMap = {
    enteros: '^[0-9]*$',
    words: '([A-z]*\\s)*',
    point25: '^\\-?[0-9]*(?:\\.25|\\.50|\\.75|)$',
    float: '/^[\\d\\.]*?(\\.\\d{0,2})?\\s*$/g',
    palabras: '/^[A-Za-z ]*$/g
  };
  constructor(private el: ElementRef) {}
  @Input('app-restrict-input')
  public set defineInputType(type: string) {
    Inputmask({regex: this.regexMap[type], placeholder: ''})
      .mask(this.el.nativeElement);
  }
}

```

Ilustración 4: Archivo restrict-input.directive.ts

```

<div class="form-group">
  <label>DESCRIPCION<a class="text-danger">*</a></label>
  <input class="form-control" type="text" app-restrict-input="palabras" style="text-transform:up
  | name="descripcion" placeholder="" required minlength="3" #descripcion="ngModel">
  <small *ngIf="descripcion.invalid && descripcion.touched" class="form-text text-danger">Ingres
</div>
<div class="form-group">
  <label> CANTIDAD </label>
  <input class="form-control" app-restrict-input="enteros" [class.is-invalid]="cantidad.invalid
  <small *ngIf="cantidad.invalid && cantidad.touched" class="form-text text-danger">Verifique es
</div>

```

Ilustración 5: Uso de la directiva

1.3. Directorio transversal

Este ataque permite tener acceso a archivos de directorios superiores directamente del equipo que se está irrumpiendo, esto se genera cuando la aplicación no está asegurada adecuadamente para realizar validaciones en entradas de texto y los parámetros de las URL's (fluidAttacks, 2018).

1.3.1. Técnica para prevenir el ataque de directorio transversal

Validar entradas en el lado del cliente mediante una función llamada sanitize

```

sanitize(string) {
  const map = {
    '&': '&amp;',
    '<': '&lt;',
    '>': '&gt;',
    '"': '&quot;',
    "'": '&#x27;',
    '/': '&#x2F;',
  };
  const reg = /[&<>"'\/]/ig;
  return string.replace(reg, (match)=>(map[match]));
}

```

Ilustración 6: Función sanitize

```

comprobadorInput(forma: NgForm){
  let resultado = false;
  for (const field in forma.controls) {
    if(forma.controls[field].value !== this.sanitize(forma.controls[field].value)){
      resultado = true;
      return resultado;
    }
  }
  return resultado;
}
alta(forma: NgForm) {
  if (this.comprobadorInput(forma)) {
    Swal.fire('Error !', 'No insertes caracteres sospechosos', 'error');
    return;
  }
}

```

Ilustración 7: Uso de la función sanitize

1.4. Intercepción criptográfica

Los ataques de intercepción criptográficos son orientados a obtener las claves utilizadas para cifrar información almacenada en un sistema (Gómez Vieites, 2014, p.6).

Según (Paguay, 2015, p.15) esta técnica intenta capturar el tráfico+ que se genera por el uso de la aplicación y así obtener datos importantes que le permitan al atacante ingresar al sistema para extraer,modificar o insertar información.

1.4.1.Técnica para prevenir Intercepción criptográfica

Encriptar las claves de autenticación

Paso 1: Instalar CryptoJS usando los siguientes comandos NPM en el directorio de su proyecto
 npm install crypto-js --save

npm install @types/crypto-js --save

Paso 2: Agregar la ruta del script en el archivo angular.json.

```

"scripts": [
  "node_modules/popper.js/dist/umd/popper.min.js",
  "node_modules/datatables.net/js/jquery.dataTables.js",
  "node_modules/jsczip/dist/jsczip.js",
  "node_modules/datatables.net-buttons/js/dataTables.buttons.js",
  "node_modules/datatables.net-buttons/js/buttons.colVis.js",
  "node_modules/datatables.net-buttons/js/buttons.flash.js",
  "node_modules/datatables.net-buttons/js/buttons.html5.js",
  "node_modules/datatables.net-buttons/js/buttons.print.js",
  "node_modules/crypto-js/crypto-js.js"
]

```

Ilustración 8: Archivo angular.json

Paso 3: Crear una clase de servicio EncrDecrService para cifrar y descifrar los métodos get/set.

```
import { Injectable } from '@angular/core';import * as CryptoJS from 'crypto-js';
@Injectable({
  providedIn: 'root'
})
export class EncrDecrService {
  constructor() { }
  set(keys, value){
    var key = CryptoJS.enc.Utf8.parse(keys);var iv = CryptoJS.enc.Utf8.parse(keys);
    var encrypted = CryptoJS.AES.encrypt(CryptoJS.enc.Utf8.parse(value.toString()), key,{
      keySize: 128 / 8,
      iv: iv,
      mode: CryptoJS.mode.CBC,
      padding: CryptoJS.pad.Pkcs7
    });return encrypted.toString();
  }
  get(keys, value){
    var key = CryptoJS.enc.Utf8.parse(keys);var iv = CryptoJS.enc.Utf8.parse(keys);
    var decrypted = CryptoJS.AES.decrypt(value, key, {
      keySize: 128 / 8,
      iv: iv,
      mode: CryptoJS.mode.CBC,
      padding: CryptoJS.pad.Pkcs7
    });return decrypted.toString(CryptoJS.enc.Utf8);
  }
}
```

Ilustración 9: clase EncrDecrService

Paso 4: Importar la clase EncrDecrService en NgModule y luego agregar este servicio en los providers.

```
FormsModule,
DataTablesModule,
ReactiveFormsModule,
],
providers: [EncrDecrService, ConfiguracionesUrl, Servicio],
bootstrap: [AppComponent]
})
export class AppModule { }
```

Ilustración 10: importación del servicio EncrDecrService

Paso 5: Utilizar los métodos de servicio cifrados y descifrados en su clase de componente.

```
this.loading = true;
this.f.password.value = this.EncrDecr.set('123456$#@1ERF', this.f.password.value);
this.authenticationService.login2(this.f.username.value, this.f.password.value).subscribe(
  data => {}
```

Ilustración 11: Encriptación de la clave de usuario

Instalar certificados SSL, con esto se encriptará el canal de comunicación cuando exista tráfico en la red

1.5. Cross-Site Request forgery

La falsificación de solicitud entre sitios (CSRF) es un ataque que obliga a un usuario final a ejecutar acciones no deseadas en una aplicación web en la que está autenticado actualmente. Con un poco de ayuda de ingeniería social un atacante puede engañar a los usuarios de una aplicación web para que ejecuten acciones de su elección. (OWASP Foundation, 2017)

1.5.1. Técnica para prevenir Cross-Site Request forgery

Instalar CSRF Protector de OWASP

Paso 1: Agregar un archivo al directorio del proyecto composer.json

```
},
"require": {
    "owasp/csrf-protector-php": "dev-master"
},
```

Ilustración 12: Archivo composer.json

Paso 2: Abrir el terminal, ir al directorio del proyecto y ejecute

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.
F:\api>compose install
```

Ilustración 13: Ejecución de compose install

Paso 3: Incluir la biblioteca y llamar a la función init()

```
header('Access-Control-Allow-Origin: *');
header("Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept");
require("conexion.php"); // IMPORTA EL ARCHIVO CON LA CONEXION A LA DB
require 'jwt_helper.php';
include_once __DIR__ . '/vendor/owasp/csrf-protector-php/libs/csrf/csrfprotector.php';
csrfProtector::init();
$conexion = conexion(); // CREA LA CONEXION
```

Ilustración 14: Uso de csrf protector

1.6. Password Mismanagement

La administración de contraseñas es un conjunto de principios y mejores prácticas que deben seguir los usuarios al almacenar y administrar contraseñas de manera eficiente para proteger las contraseñas tanto como sea posible para evitar el acceso no autorizado

1.6.1. Técnica para prevenir Password Mismanagement

Aplicar reglas de complejidad a la contraseña

```
comprobarPassword(string){  
  let contener = /^(?=.*\d)(?=.*[!@#$%^&*])(?=.*[a-z])(?=.*[A-Z]).{8,}$/;  
  return contener.test(string);  
}
```

Ilustración 15: Función que comprueba la contraseña

```
if(!this.comprobarPassword(pass1)){  
  Swal.fire("Error !", ' Las contraseñas debe tener una longitud mínima de 8. Mixtura entre mayúsculas y minúsculas, caracteres num  
  return;  
}
```

Ilustración 16: Uso de la función comprobarPassword

Encriptar las contraseñas antes de guardar

```
$opciones = [  
  'cost' => 11,  
  'salt' => mcrypt_create_iv(22, MCRYPT_DEV_URANDOM),  
];  
$db_usr_pw = password_hash($_GET['password'], PASSWORD_BCRYPT, $opciones);
```

Ilustración 17: Encriptación de la contraseña con una salt aleatoria

Combinar con una buena gestión de sesión

```
$token = array();  
$token['id'] = $db_usr_id;  
$valid_for = '1800';  
$token['exp'] = time() + $valid_for;  
array_push($datos, [ "token" => JWT::encode($token, $secret_key) ] );
```

Ilustración 18: asignación de un tiempo de expiración del token de sesión

Instalar certificados SSL, con esto se encriptará el canal de comunicación cuando exista tráfico en la red.

1.7. Privilege Escalation

La escalada de privilegios se produce cuando un usuario obtiene acceso a más recursos o funcionalidades de los que normalmente se les permiten, y la aplicación debería haber evitado dicha elevación o cambios. Esto generalmente es causado por una falla en la aplicación.

1.7.1. Técnica para prevenir Privilege Escalation

Validar ID de sesión en PHP

```
function logeado () {
    $secret_key = 'key';
    $headers = getallheaders();
    if (array_key_exists('Authorization', $headers)) {
        $jwt = $headers['Authorization'];
        $token = JWT::decode($jwt, $secret_key);
        if ($token==false){
            return 'false';
        }
        if ($token->exp >= time()) {
            return $token->id;
        } else {
            return 'false';
        }
    } else {
        return 'false';
    }
}
```

Ilustración 19: Función logeado

```
require("conexion.php"); // IMPORTA EL ARCHIVO CON LA CONEXION A LA DB
$conexion = conexion(); // CREA LA CONEXION
require 'comprobador.php';
//LLAMADO A LA FUNCION LOGEADO QUE COMPRUEBA LA SESSION
$userid = logeado();
//VERIFICA SI EL USUARIO ES CORRECTO
if ($userid != 'false') {
    ...
}
```

Ilustración 20: Uso de la función logeado

Validar sesión en Angular

```
if(localStorage.getItem('currentUser')==null || localStorage.getItem('currentUser')=='){
    this.router.navigate([''], { queryParams: { returnUrl: 'NoLogeado' }});
    return;
}
```

Ilustración 21: Comprobar si esta logeado

```
if( !(localStorage.getItem('IDPA')== '0') ){
    this.router.navigate(['template/Credito'], { queryParams: { returnUrl: 'NoPermitido' }});
}
```

Ilustración 22: Comprobar si tiene acceso al modulo

1.8. Session Fixation

La fijación de sesión es un ataque que permite a un atacante secuestrar una sesión de usuario válida. El ataque explora una limitación en la forma en que la aplicación web administra el identificador de sesión, más específicamente la aplicación web vulnerable.

1.8.1. Técnica para prevenir Session Fixation

Validar ID de sesión en PHP

```
function logeado () {
    $secret_key = 'key';
    $headers = getallheaders();
    if (array_key_exists('Authorization', $headers)) {
        $jwt = $headers['Authorization'];
        $token = JWT::decode($jwt, $secret_key);
        if ($token==false){
            return 'false';
        }
        if ($token->exp >= time()) {
            return $token->id;
        } else {
            return 'false';
        }
    } else {
        return 'false';
    }
}
```

Ilustración 23: Función logeado

```
require("conexion.php"); // IMPORTA EL ARCHIVO CON LA CONEXION A LA DB
$conexion = conexion(); // CREA LA CONEXION
require 'comprobador.php';
//LLAMADO A LA FUNCION LOGEADO QUE COMPRUEBA LA SESSION
$userid = logeado();
//VERIFICA SI EL USUARIO ES CORRECTO
if ($userid != 'false') {
```

Ilustración 24: Uso de la función logeado

Validar ID de sesión en Angular

```
comprobar() {
    let token = localStorage.getItem('token');
    let headers = new Headers();
    headers.append('Content-Type', 'application/json');
    headers.append('Authorization', ''+token);
    const requestOptions = { headers: headers };
    return this.http.get(this.URL + 'comprobar.php',requestOptions)
        .pipe(map(res => res.json()))
}
```

Ilustración 25: Función comprobar en servicios

```
comprobador() {
    this.servicio.comprobar().subscribe(
        result => {
            let ID = result.map(res => res.permiso)
            this.comprobar = ID[ID.length-1];
            if(!this.comprobar){
                this.router.navigate([''], { queryParams: { returnUrl: 'SessionCaducada' }});
            }
        }
    );
}
```

Ilustración 26: Uso de Función comprobar en modulo App-Header

BIBLIOGRAFÍA

FLUIDATTACKS (2018) *Evitar Ataques de Directorio Transversal*. Disponible en: <https://fluidattacks.com/web/es/defends/csharp/evitar-direct-transversal/> (Accedido: 6 de mayo de 2019).

GÓMEZ VIEITES, Á. (2014) *REDES INFORMÁTICAS*. Disponible en: www.keylogger.com (Accedido: 6 de mayo de 2019).

OWASP FOUNDATION (2017) «OWASP Top 10 -2017», p. 25. Disponible en:

<https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf> (Accedido: 4 de noviembre de 2018).

OWASP FOUNDATION (2018) *Falsificación de solicitudes entre sitios (CSRF) - OWASP*. Disponible en: [https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)) (Accedido: 6 de mayo de 2019).

SOTO, M. (2016) *¿Qué es XSS?* Disponible en: <https://medium.com/@marvin.soto/qué-es-xss-b9330eedbc07> (Accedido: 5 de mayo de 2019).



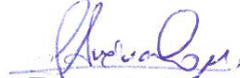
ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 10 / 08 / 2022

INFORMACIÓN DEL AUTOR	
Nombres – Apellidos:	Alex Santiago Cunachi Chacha Erick Fabricio Dávalos Mancero
INFORMACIÓN INSTITUCIONAL	
Facultad:	Informática y Electrónica
Carrera:	Software
Título a optar:	Ingeniero de Software
f. Analista de Biblioteca responsable:	 Ing. Fernanda Arévalo M.

