UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO

“SISTEMA DE INFORMACIÓN PARA EL SEGUIMIENTO Y CONTROL DEL RIESGO OPERATIVO E INCIDENTES DE SEGURIDAD” CASO: COOPERATIVA DE AHORRO Y CRÉDITO UNIÓN SANTIAGO DE MACHACA USAMA LTDA.”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Univ. Tania Ines Mamani Luque

Tutor Metodológico: Ing. Marisol Arguedas Balladares

Tutor Revisor: M. Sc. Ing. Ramiro Kantuta Limachi

Tutor Especialista: Lic. Freddy Salgueiro Trujillo

EL ALTO – BOLIVIA

2022

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

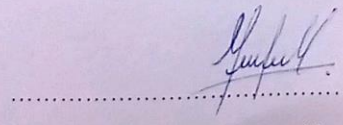
Yo **Tania Ines Mamani Luque** estudiante con **C.I. 7056228 LP.** mediante la presente declaro de manera pública que la propuesta del **TRABAJO DE GRADO** titulada **"Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad"** es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados,

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **TRABAJO DE GRADO** haya sido publicado anteriormente, asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, Noviembre de 2022



.....

Tania Ines Mamani Luque
C.I. 7056228 LP.
e-mail: tania.ines.15@gmail.com

“Para empezar un proyecto, hace falta valentía. Para terminar un proyecto, hace falta perseverancia”.

DEDICATORIA

Dedico este trabajo principalmente a Dios, por permitirme haber llegado hasta este momento tan importante de mi formación profesional. A mis padres, por su amor, trabajo y sacrificio en todos estos años. A mi hermana por estar siempre presente, acompañándome en todo momento. A todas las personas que me apoyaron y han hecho que el trabajo se realice con éxito, en especial a aquellos que me abrieron sus puertas y compartieron sus conocimientos.

Tania Ines Mamani Luque

AGRADECIMIENTO

A Dios, por darme la vida y estar siempre conmigo, guiándome en mí camino.

A mis padres: Mario Antonio Mamani Callisaya y Virginia Luque de Mamani. Por darme la vida, su amor, su paciencia, su apoyo incondicional y por creer en mí.

A mi Hermana: Yesica Mamani Luque por estar conmigo en todo momento y brindarme palabras de aliento, gracias.

Agradecer a todos los que tuvieron que ver en el desarrollo y conclusión de este proyecto de grado.

A mi tutor metodológico Ing. Marisol Arguedas Balladares por su conocimiento, apoyo, confianza, tiempo, paciencia y motivación que brindo hacia a mi persona. Agradezco que me haya dado la oportunidad de ser su estudiante.

A mi tutor especialista Lic. Freddy Salgueiro Trujillo por compartir sus conocimientos, brindarme orientación, sugerencias con paciencia motivación durante el desarrollo del presente proyecto.

A mi tutor revisor Ing. Ramiro Kantuta Limachi por su disponibilidad de tiempo, su acertada orientación y observaciones brindadas, por compartir sus conocimientos, brindarme su ayuda, sugerencias con paciencia motivación durante el desarrollo del presente proyecto.

A la Universidad Pública de El Alto, por acogerme en sus aulas durante todos los años de estudio, así también a la Carrera Ingeniería de Sistemas y a mis compañeros (as)de estudio por su apoyo incondicional.

INDICE GENERAL

Pág.

ÍNDICE DE FIGURAS

Pág.

ÍNDICE DE TABLAS

Pág.

RESUMEN

El presente proyecto de grado plasma el desarrollo del Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. de la ciudad de El Alto, el cual está basado en un enfoque de procesos que se adaptan a su realidad y necesidades actuales, con el objetivo de mejorar la operatividad dentro de la Institución. Por tanto, primero es necesario modelar detalladamente los procesos de la Cooperativa y sobre ello diseñar y construir el Sistema de Información.

El desarrolló del sistema será de importancia en el manejo de la información, es así que se tiene como objetivo el desarrollar un Sistema de Información que ayude al Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad, para cumplir con lo que establece la ASFI. El resultado de este proyecto, será lograr que la Cooperativa centralice toda información en una base de datos y obtener del mismo los reportes correspondientes a la Central de Riesgo Operativo.

El análisis y diseño del sistema se desarrolló con la metodología RUP, para evaluar la calidad del software se utilizó la ISO/IEC 9126 que permiten conocer el nivel de la calidad del software a través de un proceso de evaluación de acuerdo con las métricas o indicadores que presenta el modelo de calidad, en seguridad de la información se recurrió a la norma ISO 27001 y finalmente para la estimación de costos se usó COCOMO II.

Finalmente se concluye que se lograron cumplir con los objetivos que fueron planteados y que el sistema fue implementado dentro de la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

Palabras clave: Sistema, Seguimiento, Control, Riesgo, Incidente, RUP.

ABSTRACT

This degree project reflects the development of the Information System for the Monitoring and Control of Operational Risk and Security Incidents in the Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. of the city of El Alto, which is based on a process approach that adapts to your reality and current needs, with the aim of improving the operation within the Institution. Therefore, first it is necessary to model in detail the processes of the Cooperative and on this design and build the Information System.

The development of the system will be of importance in the management of information, so the objective is to develop an Information System that helps the Monitoring and Control of Operational Risk and Security Incidents, to comply with what is established by ASFI. The result of this project will be to ensure that the Cooperative centralizes all information in a database and obtains from it the corresponding reports to the Operational Risk Center.

The analysis and design of the system was developed with the RUP methodology, to evaluate the quality of the software the ISO/IEC 9126 was used, which allows knowing the level of software quality through an evaluation process according to the metrics or indicators. that presents the quality model, in information security the ISO 27001 standard was used and finally for the cost estimation COCOMO II was used.

Finally, it is concluded that the objectives that were set were met and that the system was implemented within the Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

Keywords: System, Monitoring, Control, Risk, Incident, RUP.

CAPÍTULO I

CAPÍTULO I
MARCO PRELIMINAR

1.1. INTRODUCCIÓN

El entorno que rodea a las Entidades Financieras está cambiando rápidamente expandiendo los riesgos a los que ésta se enfrenta, el Riesgo Operacional[1] no es un hecho nuevo, la gestión de este riesgo ha cobrado importancia en el último tiempo en las instituciones financieras. Consecuentemente, las organizaciones reguladoras a nivel mundial han integrado criterios para la revisión de riesgos, tomando como base, los Lineamientos del Comité de Supervisión Bancaria de Basilea[2], tal es el caso de la Autoridad de Supervisión del Sistema Financiero ASFI[3].

Por otro parte la información es uno de los activos más valiosos que posee las Entidades Financieras, debido a su importancia se requiere establecer un conjunto de medidas que permitan la gestión de los riesgos, la preservación de los activos y la continuidad del negocio. Un sistema para la gestión de Incidentes de Seguridad de la Información resulta fundamental para tener la capacidad de gestionar los defectos que se vayan presentando en las Instituciones Financieras.

La Cooperativa anualmente debe atravesar por Inspecciones por parte de la Entidad reguladora ASFI., mismo que en cada visita buscan el cumplimiento de los requisitos mínimos para la obtención de la Licencia de Funcionamiento, desde el año 2019 la Cooperativa arrastra las siguientes observaciones*: "No tiene desarrollado… Artículo 1, Sección 6, Capítulo II, Título V, Libro 3"* (Mendoza Tellez, 2020)*, "No cuenta con los reportes…Articulo 2, Sección 6, Capítulo II, Título V, Libro 3"* (Mendoza Tellez, 2020)*, "La entidad no efectuó un diagnóstico…Articulo 2, Sección 2, Capítulo III, Titulo*

[1] El Comité de Basilea define riesgo operacional como el riesgo de incurrir en pérdidas debido a la inadecuación o a fallos de los procesos, el personal y los sistemas internos o bien a causa de acontecimientos externos. Esta definición incluye el riesgo legal, pero excluye el riesgo estratégico y reputación.

[2] El Comité de Supervisión Bancaria de Basilea es un comité de autoridades de supervisión bancaria que fue establecido por los gobernadores de los bancos centrales de los países del Grupo de los Diez en 1975. Suele reunirse en el Banco de Pagos Internacionales de Basilea, donde se encuentra su Secretaría permanente.

[3] ASFI. es una Institución de derecho público y de duración indefinida, con personalidad jurídica, patrimonio propio y autonomía de gestión administrativa, financiera, legal y técnica, con jurisdicción, competencia y estructura de alcance nacional, bajo tuición del Ministerio de Economía y Finanzas Públicas y sujeta a control social.

VII, Libro 3" (Mendoza Tellez, 2020), tras la inexistencia de un sistema la Cooperativa USAMA Ltda. maneja los eventos de riesgo operativo e incidentes de Seguridad de la Información en formularios los cuales causan una demora en los tiempos provocando tardanzas para el Encargado de Riesgos y el Oficial de Seguridad de la Información, asimismo de existir retrasos en los reportes trimestrales que deben ser remitidos a través del SCIP[4] y los informes que deben ser entregados a la Alta Gerencia y a la ASFI., mostrando de ese modo que no existe capacidad de envió de información a la ASFI., sin contar que podrían existir pérdidas que afecten el estado de ganancias y pérdidas por tales demoras.

El proyecto utilizará la metodología RUP para realizar las fases de inicio, elaboración, construcción y transición del mismo, el cual permitirá el Seguimiento y Control del sistema, lo cual conlleva a tener una información más precisa por el Encargado de Riesgos y el Oficial de Seguridad de la Información, seguimiento a los eventos de riesgo operativo[5] registrados, incidentes de Seguridad de la Información[6] y la validación a través de SCIP dentro de la Cooperativa USAMA Ltda., toda la información que se obtenga se centralizará en una base de datos, que contribuirá a la automatización de la misma. Además, se hará el uso de la herramienta UML (Lenguaje Unificado de Modelo) que permite visualizar los eventos e incidentes de riesgo que registran los funcionarios en la Entidad.

1.2. ANTECEDENTES

1.2.1. Antecedentes institucionales

La Cooperativa de Ahorro y Crédito Unión Santiago de Machaca "USAMA" Ltda., nace como una Cooperativa de Consumo con Sección de Ahorro y Crédito, a raíz de una gran preocupación e inquietud de un grupo de personas denominada

[4] Sistema de Captura de Información Periódica (SCIP): Sistema que provee a las entidades supervisadas la estructura y los mecanismos necesarios para capturar y aplicar validaciones mínimas de formato y consistencia de los datos, previo al envío de la información periódica.

[5] Evento de riesgo operativo: Es un incidente o conjunto de ellos, que provocan que los resultados difieran de los esperados, debido a procesos defectuosos, recursos humanos inadecuados, fallos en los sistemas o por causas externas.

[6] Incidente de seguridad de la información: Suceso o serie de sucesos, que tienen una probabilidad significativa de comprometer las operaciones de la Entidad Supervisada, amenazar la seguridad de la información y/o los recursos tecnológicos.

Acción Familiar Villalobos "AFAVI" por la aguda crisis económica que atravesó Bolivia en los '80. Bajo ese contexto de agobiante situación, económica y social, surge la idea de organizarse y formar la Cooperativa, a la cabeza del hermano Marcos Villalobos Alcón en marzo de 1984. Este gran esfuerzo dio lugar a la creación de la Cooperativa a través del Centro de Acción Social "Unión Santiago de Machaca", fundado el 25 de diciembre de 1954, con Personería Jurídica No. 70095.

La Cooperativa Unión Santiago de Machaca “USAMA Ltda.[7]”, fue fundada el 19 de julio de 1985, cuenta con Personería Jurídica reconocida mediante Resolución de Consejo No. 03026 de 28 de mayo de 1986 e inscrita en el Registro Nacional de Cooperativas con el número 2773 de la misma fecha. La Cooperativa cuenta desde entonces con más de 35 años de vigencia. En aplicación de la Ley General de Sociedades Cooperativas, mediante Resolución del Consejo Nacional de Cooperativas, el Estatuto orgánico se registró con el número 4216, con el nombre de Cooperativa de Ahorro y Crédito Unión Santiago de Machaca “USAMA Ltda.”, también constituyen el marco normativo de la entidad, desde junio de 2017 se recibió el Certificado de Adecuación extendido por la Autoridad de Supervisión del Sistema Financiero (ASFI.), el cual permite el desarrollo legal de las actividades de la Cooperativa, bajo sanas prácticas de intermediación financiera.

La Oficina Central se encuentra ubicada en la avenida Juan Pablo II No. 2805 de la Zona de Ferropetrol, altura Cruz Papal lado Cristembo de la ciudad de El Alto, la Oficina Externa, ubicado en la calle Isaac Tamayo No. 662, Galería Isaac Tamayo, zona El Rosario del centro de la ciudad de La Paz y de una Oficina Ferial, instalada en la plaza principal de la localidad de Santiago de Machaca en ambientes de la sub gobernación de la Provincia José Manuel Pando, Departamento de La Paz.

Misión

“Ofertar servicios financieros integrales al mercado de personas, operando con ética y orientados al crecimiento de la entidad, los socios, los empleados y la Comunidad” así mismo, contribuir al desarrollo y mejora de la calidad de vida de

[7] Ltda.: Sociedad de Responsabilidad Limitada es una persona jurídica o empresa formada por un mínimo de dos y un máximo de cincuenta socios, quienes limitan su responsabilidad al monto aportado como capital.

nuestros socios y usuarios, esperando con ética y orientados al crecimiento de la entidad, socios, colaboradores y la comunidad.

Visión

"Ser la entidad líder en el mercado de personas, ofertando servicios financieros integrales, buscando negocios sostenibles a largo plazo, así como el bienestar de los grupos de interés y la comunidad".

Figura 1

Estructura Orgánica 2020 Cooperativa de Ahorro y Crédito Unión Santiago de Machaca Ltda.

Nota. Estructura Orgánica 2020, donde se encuentran las áreas que presentan problemas e inconvenientes dentro de la Cooperativa USAMA Ltda.

1.2.2. Antecedentes internacionales

Tabla 1

Primer estado de arte internacional

Antecedente Internacional	
Título de la publicación	"Administración del riesgo operativo y los efectos económicos-financieros antes y posterior a la normativa de la gestión del riesgo operativo".

Lugar	País: Ambato-Ecuador Año: 2018
Autor/es	Muncha Quezada, Ana Cristina
Palabras claves	Administración del Riesgo Operativo, Efectos Económicos-Financieros, Normativa, Gestión.
Dirección electrónica específica	http://repositorio.uta.edu.ec/jspui/handle/123456789/28534

TÓPICOS RELEVANTES

Formulación del problema o preguntas de investigación	Estudios previos han destacado algunas de las características del riesgo operacional, su vida diaria y lentitud, sus especificidades según los tipos de acción, los eventos y las innovaciones introducidas por el regulador. ¿Incide el desarrollo de la administración del riesgo operativo en los efectos económicos-financieros de las cooperativas de ahorro y crédito de la ciudad de Ambato?
Objetivo de la investigación	Estudiar el proceso de la administración del riesgo operativo y sus efectos económicos-financieros. (Muncha Quezada, 2018)
Hipótesis/Idea a defender	El proceso de administración del riesgo operacional incide en los efectos económicos financieros de las cooperativas de ahorro y crédito de la ciudad de Ambato.
Metodología de la investigación o desarrollo	La metodología de investigación que se uso fue: Investigación bibliográfica documental e Investigación de campo.
Resultados de la investigación	Luego de comparar el alcance y naturaleza del riesgo operacional antes y posterior a la normativa de Basilea, uno de los mayores avances que se consiguió fue el consensuar una definición para riesgo operacional y es que

	a partir de ella se pudo establecer una clasificación y determinación de los principales factores y eventos de pérdida.
ANALISIS DEL TRABAJO	
Discusión	En relación al proyecto de la Carrera de Ingeniería Financiera al solo ser de investigación se pudo destacar que pudo llegar a cumplir sus objetivos (obtener una clasificación de los factores de pérdida) resulta ser interesante considerando la amplia población que tuvo que estudiar (todas las cooperativas de ahorro y crédito).

Nota. Primer antecedente internacional, extraído del Proyecto realizado por Ana Cristina Muncha Quezada, 2018.

Tabla 2

Segundo estado de arte internacional

Antecedente Internacional	
Título de la publicación	“Riesgo Operativo y su influencia en las decisiones financieras en las Cajas de Ahorro y Crédito en Lima provincia, año 2018”.
Lugar	País: Lima-Perú Año: 2020
Autor/es	Bonifacio Carrasco Kuinny, Velásquez Rojas Javier Alexander
Palabras claves	Riesgo operativo, decisiones financieras.
Dirección electrónica específica	https://repositorio.usmp.edu.pe/handle/20.500.12727/7590

TÓPICOS RELEVANTES

Formulación del problema o preguntas de investigación	El Sistema de CMAC's en el Perú se han posicionado, como elemento fundamental para descentralización y democratización del crédito en el país, cuyo objetivo principal es la de evitar la exclusión social además de fomentar el desarrollo económico y el progreso social de sus comunidades de origen. ¿De qué manera el riesgo operativo influye en las decisiones financieras de las cajas de ahorro y crédito de lima provincia en el año 2018?
Objetivo de la investigación	Determinar la influencia existente del riesgo operativo en las decisiones financieras de las cajas de ahorro y crédito en lima provincia en el año 2018. (Bonifacio Carrasco & Velasquez Rojas, 2020)
Hipótesis/Idea a defender	El riesgo operativo influye de manera significativa en las decisiones financieras de las cajas de ahorro y crédito de lima provincia en el año 2018.
Metodología de la investigación o desarrollo	Es una investigación no experimental, cuyo diseño metodológico es el transaccional correlacional. Ox r Oy. Dónde: O = Observación, x = Riesgo operativo, y = Decisiones financieras, r = Relación de variables.

Resultados de la investigación	En las cajas de ahorro crédito de Lima Provincia, los sistemas de información influyen de manera positiva en el desarrollo de la información financiera, así también pueden optimizar el buen desarrollo financiero y nos ayudará en la mitigación del riesgo operativo existente. Así también, afirmamos que la información financiera de la empresa es importante al momento de evaluar y tomar decisiones de carácter económico, además de mejorar las medidas adoptadas en el plan operativo, todo esto basado en los resultados obtenidos y en el estudio realizado.
ANALISIS DEL TRABAJO	
Discusión	En relación al proyecto de la Facultad de Contabilidad y Finanzas se pudo resaltar que los investigadores destacan la manera positiva la existencia de un sistema de información en común el cual ayuda a mitigar los riesgos existentes en las cajas de ahorro y crédito de Lima provincia.

Nota. Segundo antecedente internacional, extraído del Proyecto realizado por Kuinny Bonifacio Carrasco, Javier Alexander Velásquez Rojas, 2020.

Tabla 3

Tercer estado de arte internacional

Antecedente Internacional		
Título de la publicación	“Sistema de Gestión de Incidencias”.	
Lugar	País: Madrid-España	Año: 2017
Autor/es	Ambrós Mendioroz, Miguel	
Palabras claves	Sin palabras clave	

Dirección electrónica específica	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi_2veOmoz6AhVqlJUCHYTTA8oQFnoECBMQAQ&url=https%3A%2F%2Foa.upm.es%2F47458%2F3%2FTFC_MIGUEL_AMBROS_MENDIORORZ.pdf&usg=AOvVaw0gRoSwO5e1v99xWnaXZ56

TÓPICOS RELEVANTES

Formulación del problema o preguntas de investigación	Para realizar un correcto control de las incidencias que vayan surgiendo en las distintas aplicaciones que se desarrolla dentro del departamento, cada vez más numerosas y complejas, se observa la necesidad de disponer de una herramienta que facilite estas tareas.
Objetivo de la investigación	El objetivo principal como se ha comentado anteriormente es desarrollar una aplicación web para realizar la gestión de incidencias. Este desarrollo debe realizarse de la forma más rápida posible. Esta aplicación debe permitir realizar un seguimiento sencillo de las incidencias detectadas, de forma acorde a las necesidades de los departamentos de desarrollo y pruebas. (Ambros Mendioroz, 2017)
Hipótesis/Idea a defender	Sin hipótesis
Metodología de la investigación	De las distintas metodologías agiles disponibles la elegida es SCRUM.

Resultados de la investigación	Como se comentó al inicio, el objetivo de este trabajo era el diseñar y construir una aplicación web para facilitar la gestión de incidencias en una determinada empresa, que hasta el momento estaba utilizando unos sistemas muy básicos para realizar dicha gestión. Este objetivo se encuentra alcanzado, ya que la aplicación desarrollada supone una importante mejora con respecto a los métodos que se utilizaba inicialmente.
ANALISIS DEL TRABAJO	
Discusión	En relación al proyecto (aplicación web) de la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, se resalta de manera positiva el haber utilizado una metodología acorde a las circunstancias por tratarse de una aplicación que fue realizado en poco tiempo y adaptándose a cambios de última hora los cuales fueron solicitados por una compañía de comercio minorista.

Nota. Tercer antecedente internacional, extraído del Proyecto realizado por Miguel Ambrós Mendioroz, 2017.

1.2.3. Antecedentes nacionales

Tabla 4

Primer estado de arte nacional

Antecedente Nacional	
Título de la publicación	"Guía metodológica de Gestión de Riesgo Operacional para Instituciones Financieras de desarrollo de Bolivia".
Lugar	País: La Paz-Bolivia Año: 2017
Autor/es	Aramayo Shaw, Ingrid Andrea
Palabras claves	Sin palabras clave

Dirección electrónica específica	http://repositorio.uasb.edu.bo:8080/handle/54000/517

TÓPICOS RELEVANTES

Formulación del problema o preguntas de investigación	Las directrices emitidas por ASFI., brindan pautas básicas que permiten a las Instituciones Financieras de Desarrollo definir su modelo de gestión de riesgo operacional, pero debido a la escaza experiencia que se tiene sobre el riesgo operacional y su gestión, ha generado que los modelos implementados por las IFD´s no contribuyan en gran medida a la correcta y oportuna apreciación y tratamiento del riesgo operacional. ¿De qué manera una Institución Financiera de Desarrollo de Bolivia puede apreciar y tratar los riesgos operacionales a los cuales se encuentra expuesta?
Objetivo de la investigación	Diseñar una Guía Metodológica de Gestión de Riesgo Operacional que permita a una Institución Financiera de Desarrollo de Bolivia apreciar y tratar el riesgo operacional al cual se encuentra expuesta.(Aramayo Shaw, 2017)
Hipótesis/Idea a defender	La guía metodología de gestión del riesgo operacional permitirá a las Instituciones Financieras de Desarrollo de Bolivia apreciar y tratar el riesgo operacional al cual se encuentran expuestas.

Metodología de la investigación o desarrollo	En el presente trabajo de investigación se empleó el método deductivo, este método parte lo general hasta llegar a lo específico. El método deductivo comienza dando paso a los datos en cierta forma válidos, para llegar a una deducción a partir de un razonamiento de forma lógica o suposiciones; o sea se refiere a un proceso donde existen determinadas reglas y procesos donde gracias a su asistencia, se llegan a conclusiones finales partiendo de ciertos enunciados o premisas.
Resultados de la investigación	Finalizada la investigación sobre la situación actual de las IFD´s en relación a la Gestión del Riesgo Operacional se establece que estas Instituciones se encuentran en un proceso de adecuación considerando que obtuvieron su Licencia de Funcionamiento a finales de la gestión 2016, es importante que dentro de este proceso consideren los aspectos mínimos establecidos por ASFI. en relación a la Gestión del Riesgo Operacional con el objeto de dar cumplimiento a este marco regulatorio lo cual les permitirá implementar un Sistema de Gestión de Riesgo Operacional acorde a los lineamientos establecidos por ASFI.

ANALISIS DEL TRABAJO

Discusión	En relación a la tesis del Área de Finanzas y Proyectos Empresariales, se resalta Aramayo Shaw la atribución que hizo al ampliar las pautas que adicionó con su trabajo sobre el riesgo operacional más allá de las

	emitidas por la ASFI. para el beneficio de las instituciones financieras de desarrollo.

Nota. Primer antecedente nacional, extraído de la Tesis realizado por Ingrid Andrea Aramayo Shaw, 2017.

Tabla 5

Segundo estado de arte nacional

Antecedente Nacional	
Título de la publicación	"Riesgo Operativo en el proceso de emisión de giros al exterior".
Lugar	País: La Paz-Bolivia Año: 2019
Autor/es	Deheza Salinas, Aneth Stefany
Palabras claves	Sin palabras clave
Dirección electrónica específica	https://repositorio.umsa.bo/handle/123456789/21420
TÓPICOS RELEVANTES	
Formulación del problema o preguntas de investigación	Habiendo realizado un análisis inicial del Banco Nacional de Bolivia S.A., se pudo evidenciar que esta entidad dispone de normativa interna que mitiga el riesgo operativo al cual se expone al momento de efectuar transferencias al exterior de clientes. Sin embargo, el escenario actual donde la entidad desenvuelve sus actividades, debido a los fraudes tanto internos y externos a los que están expuestos los bancos, se ven en la necesidad de reforzar sus mecanismos de seguridad a fin de mitigar el riesgo operativo.

Objetivo de la investigación	El objetivo que persigue el trabajo es identificar los factores inherentes al riesgo operativo que contiene el procesamiento de un giro al exterior a solicitud de sus clientes a los cuales se expone el Banco Nacional de Bolivia S.A. (Deheza Salinas, 2019)
Hipótesis/Idea a defender	Sin hipótesis
Metodología de la investigación o desarrollo	Se utilizará la metodología descriptiva al fin de llegar a conocer los escenarios predominantes que conlleva la realización de un giro al exterior a través de la descripción exacta de procedimientos, actividades y personas involucradas, sin la delimitación a la recolección de datos, sino más bien a edificación de la relación que existe entre dos o más variables.
Resultados de la investigación	El Banco Nacional de Bolivia S.A. cuenta con marco normativo elaborado y aprobado por las instancias pertinentes, de acuerdo a atributos y responsabilidades asignados según su estructura organizacional. En el proceso de Emisión de giros al exterior participan un total de hasta de 3 a más personas, de acuerdo a la naturaleza del giro, es decir que mientras mayor sea el monto mayor participación de personas conlleva su realización. El BNB cuenta con un sistema informático pertinente para el procesamiento de giros al exterior.

ANALISIS DEL TRABAJO	
Discusión	En relación a la monografía realizada en la Facultad de Ciencias Económicas y Financieras, se resalta que el riesgo operativo se encuentra en todo sitio e institución y en este caso el BNB el cual ya cuenta con un sistema informático acorde a los giros y una estructura organizada pertinente.

Nota. Segundo antecedente nacional, extraído de la Monografía realizado por Aneth Stefany Deheza Salinas, 2019.

Tabla 6

Tercer estado de arte nacional

Antecedente Nacional	
Título de la publicación	"Modelo de seguridad basado en la norma ISO/IEC 27001/2013, para minimizar los riesgos en la seguridad lógica de la información"
Lugar	País: Santa Cruz-Bolivia Año: 2019
Autor/es	Akly Aguayo, Fabiola
Palabras claves	Sin palabras clave
Dirección electrónica específica	https://www.soe.uagrm.edu.bo/books/modelo-de-seguridad-basado-en-la-norma-iso-iec-27001-2013-para-minimizar-los-riesgos-en-la-seguridad-logica-de-la-informacion/
TÓPICOS RELEVANTES	
Formulación del problema o preguntas de investigación	Para minimizar los riesgos relacionados a la seguridad lógica de la información: seguridad de contraseñas, código malicioso, manipulación de la información, privilegios de Capacitaciones y amenazas ambientales y naturales. ¿Cómo contribuir

	a la seguridad lógica de la información aplicando gestión de riesgo en la Dirección de Gestión Catastral?
Objetivo de la investigación	Desarrollar un Modelo de Seguridad, basado en la ISO /IEC 27001/2013, para minimizar los riesgos de la seguridad lógica de la información en la Dirección de Gestión Catastral. (Akly Aguayo, 2019)
Hipótesis/Idea a defender	La aplicación de un Modelo de Seguridad, basado en la ISO /IEC 27001/2013, minimiza los riesgos de la seguridad lógica de la información en la Dirección de Gestión Catastral.
Metodología de la investigación o desarrollo	Metodología de Análisis y Gestión de Riesgo Magerit: Es una metodología de análisis y gestión de riesgo, elaborada por el consejo Superior de Administración Electrónica, en respuesta a la necesidad de que toda la sociedad, organizaciones que dependen de las tecnologías de información para el cumplimiento de su misión.
Resultados de la investigación	Con un enfoque histórico lógico y a partir de la sistematización a diferentes autores se fundamentaron teóricamente los aspectos relacionados con la gestión de riesgo, identificación, análisis, tratamiento y evaluación del mismo, unido a la normativa referida a la gestión de riesgo. Así también, se analizaron los temas que identifican la seguridad de la información.

ANALISIS DEL TRABAJO	
Discusión	En relación a la Tesis de la Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones, se pudo destacar que pudo llegar a cumplir sus objetivos (contribuir con la seguridad) creando un modelo de seguridad basado en la ISO/IEC 27001 en la Dirección de Gestión Catastral.

Nota. Tercer antecedente nacional, extraído de la Tesis realizado por Fabiola Akly Aguayo, 2019.

1.2.4. Antecedentes locales

Tabla 7

Primer estado de arte local

Antecedente Local	
Título de la publicación	"Gestión de Riesgo Operativo[8] para el Banco FIE S.A. sobre la base de Normativa Vigente de la ASFI."
Lugar	País: La Paz-Bolivia Año: 2017
Autor/es	Quispe Nina, Jhovana Wendy
Palabras claves	Riesgo, Tipo de riesgo, Operativo, Tipo de evento de pérdida.
Dirección electrónica específica	https://repositorio.umsa.bo/handle/123456789/20349
TÓPICOS RELEVANTES	
Formulación del problema o	La identificación, medición, control, mitigación y difusión de los riesgos se la efectuara a partir del registro de eventos de riesgo operativo, hallazgos de

[8] Gestión de riesgo operativo: Es el proceso estructurado, consistente y continuo para identificar, medir, monitorear, controlar, mitigar y divulgar el riesgo operativo al cual la entidad supervisada se encuentra expuesta en el marco del conjunto de estrategias, objetivos, políticas, procedimientos y acciones, establecidas por la entidad para este propósito.

preguntas de investigación	la unidad de Auditoría Interna, de Auditorías Externas o inspecciones de ASFI., informes de la Gerencia Nacional de Seguridad de la Información y Continuidad de Negocio, los análisis de casos a requerimiento de o a través de Directorio y Comité de Gestión Integral de Riesgos, la autoevaluación de procesos y las evaluaciones de productos nuevos.
Objetivo de la investigación	Proponer una Gestión de Riesgo Operativo Adecuada para el Banco FIE S.A. sobre la base de Normativa Vigente de la (ASFI.) considerando los riesgos que asume logrando el cumplimiento de metas y objetivos propuestos. (Quispe Nina, 2017)
Hipótesis/Idea a defender	Sin hipótesis
Metodología de la investigación o desarrollo	INDUCTIVO: De hechos particulares sacamos una conclusión general y DEDUCTIVO: De normas preestablecidas se sacarán conclusiones.
Resultados de la investigación	La elección de un determinado modelo para medir el riesgo, sea de mercado o de crédito, es una elección subjetiva, apoyada en un juicio. Los procedimientos nuevos y actualizados contaran con una autoevaluación por parte de los dueños de procesos, donde se identifiquen los riesgos operativos a los que se expone Banco FIE, de tal forma que se tenga clara las actividades de control interno que mitigan dicha exposición.

ANALISIS DEL TRABAJO	
Discusión	En relación al proyecto realizado en la Facultad de Ciencias Económicas y Financieras, se resalta que el trabajo realizado fue para evitar observaciones por parte del ente regulador, anticipando y proponiendo una guía de Gestión de Riesgo Operativo Adecuada para todas las agencias del Banco FIE S.A. en Bolivia.

Nota. Primer antecedente local, extraído del Proyecto realizado por Jhovana Wendy Quispe Nina, 2017.

Tabla 8

Segundo estado de arte local

Antecedente Local	
Título de la publicación	"Gestión de riesgo operativo en las instituciones Microfinancieras de Bolivia en el período 2006-2018".
Lugar	País: La Paz- Bolivia Año: 2022
Autor/es	Uturunco Mamani, Wendy Nicole
Palabras claves	Sin palabras clave.
Dirección electrónica específica	https://repositorio.umsa.bo/handle/123456789/27911
TÓPICOS RELEVANTES	
Formulación del problema o preguntas de investigación	El riesgo Operativo es la probabilidad de incurrir en pérdida como en consecuencia de deficiencias o fallos de los procesos interno, errores humanos, mal funcionamiento de los sistemas de información y eventos externos. El objetivo principal de la gestión de riesgo operativo es garantizar la identificación y administración optima del riesgo de forma rentable y

	que sea congruente con el apetito de riesgo de la institución.
Objetivo de la investigación	Analizar los principales elementos determinantes del riesgo operativo de las entidades de intermediación financiera con especialización en microfinanzas de Bolivia del 2006-2018. (Uturunco Mamani, 2022)
Hipótesis/Idea a defender	La previsión genérica para incobrabilidad de cartera por factores de riesgo adicional, fraude interno por apropiación ilegal de activos, fallas en hardware y software y seguros por eventos externos son determinantes del proceso de gestión de riesgo operativo en las entidades de intermediación financiera con especialización en microfinanzas de Bolivia del 2006-2018.
Metodología de la investigación o desarrollo	La investigación, emplea el método de investigación deductivo, esta aproximación se vale de la lógica o razonamiento deductivo, que comienza con la teoría, y de esta se derivan expresiones lógicas denominadas hipótesis que el investigador somete a prueba.

Resultados de la investigación	Se logra observar por medio de la presente tesis que la gestión de riesgo operativo en las entidades de intermediación financiera con especialización en microfinanzas es deficiente, si apelamos a la estimación del modelo econométrico de base de datos de panel (efectos fijos) las variables que aumentan más el riesgo operativo son: el fraude interno por apropiación ilegal de activos en 0.47% y los fallos de hardware y software en 0.11% concretizando así que es deficiente.
ANALISIS DEL TRABAJO	
Discusión	En relación a la Tesis realizado en la Carrera de Economía, se resalta que cumplió con sus objetivos (analizar los elementos determinantes del riesgo operativo aplicando un modelo econométrico de base de datos) resulta ser atrayente considerando la amplia población que tuvo que estudiar (todas las instituciones Microfinancieras en Bolivia).

Nota. Segundo antecedente local, extraído de la Tesis realizado por Wendy Nicole Uturunco Mamani, 2022.

Tabla 9

Tercer estado de arte local

Antecedente Local	
Título de la publicación	"Plan de implementación de Gobierno Electrónico 2017-2025".
Lugar	País: La Paz- Bolivia Año: 2017
Autor/es	Comité Plurinacional de Tecnologías de la Información y Comunicación COPLUTIC y Agencia

	de Gobierno Electrónico y Tecnologías de la Información y Comunicación AGETIC.
Palabras claves	Sin palabras clave.
Dirección electrónica específica	https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwi92ZjQr4z6AhU0FLkGHXmDDtEQFnoECAcQAQ&url=https%3A%2F%2Fcoplutic.gob.bo%2FIMG%2Fpdf%2Fplan_gobierno_electronico_.pdf&usg=AOvVaw1q9sQwg4L7sB0oWW9mmKex

TÓPICOS RELEVANTES

Formulación del problema o preguntas de investigación	El uso de tecnologías de la información y comunicación (TIC) para fortalecer las capacidades e incrementar la eficiencia de las entidades públicas, como para mejorar los canales de comunicación del Estado con la ciudadanía y la participación y control social, proviene de varios años atrás. Sin embargo, las iniciativas han sido dispersas y han quedado obsoletas debido a la construcción del Estado y al impulso social que transforma la estructura económica, social y política del país.
Objetivo de la investigación	Modernizar y transparentar la gestión pública, otorgando servicios y atención de calidad a la ciudadanía, garantizando el derecho a la información, así como contribuir a la eficiencia y eficacia de la actividad administrativa en los procesos internos del Gobierno, mediante el uso de las tecnologías de información y comunicación y otras herramientas. (COPLUTIC y AGETIC, 2017)

Hipótesis/Idea a defender	Sin hipótesis
Resultados de la investigación	La AGETIC es la entidad responsable de evaluar y realizar el seguimiento a la calidad y eficiencia de los servicios de Gobierno Electrónico y monitorear el Plan de Implementación de Gobierno Electrónico por tanto, el seguimiento y evaluación se centrarán en el cumplimiento de metas, resultados, líneas estratégicas definidas en el marco de los ejes: Gobierno Soberano, Gobierno Eficiente y Gobierno Abierto y Participativo. Seguimiento constante, Evaluaciones parciales (anuales), Evaluación de medio término, Evaluación final del plan.
ANALISIS DEL TRABAJO	
Discusión	En relación al trabajo realizado por las Instituciones COPLUTIC y la AGETIC, se logra destacar que acorde a la Ley Nº 164 crea lineamientos para mejorar los canales de comunicación del Estado con la ciudadanía, implementando mecanismos para evitar incidentes informáticos.

Nota. Tercer antecedente local, extraído del Trabajo realizado por el Comité Plurinacional de Tecnologías de la Información y Comunicación "COPLUTIC" y la Agencia de Gobierno Electrónico y Tecnologías de la Información y Comunicación AGETIC, 2017.

1.3. PLANTEAMIENTO DEL PROBLEMA

El riesgo es una pieza presente en todas las actividades humanas y es un factor principal a la hora de tomar cualquier tipo de decisión. Este término hace referencia a la cercanía o posibilidad de daño o pérdida, que han afectado el patrimonio y reputación de algunas entidades financieras causadas por factores de Riesgo Operativo. La necesidad de implementar cada vez más sistemas de información genera un problema para los administradores de sistemas de las entidades públicas,

el cual hace que descuiden las configuraciones de seguridad de las aplicaciones, servicios e infraestructuras informáticas, lo que provoca que sean vulnerables a ataques cibernéticos.

Actualmente la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca Ltda. cuenta con los lineamientos para la gestión de Riesgo Operativo los cuales se encuentran establecidos en el Manual de Políticas y Normas de Gestión de Riesgo Operativo y el procedimiento de Incidentes de Seguridad de la Información que se encuentra publicado en el Portal Interno USAMANET dentro del Marco Formal de la Entidad. La Cooperativa USAMA Ltda. recibe Inspecciones cada año por la Entidad de regulación ASFI. dentro de sus instalaciones con el objetivo de presentar y cumplir los requisitos mínimos para la obtención de la Licencia de Funcionamiento, tras cada visita estos últimos años, la Cooperativa mantuvo sus observaciones.

Tras las observaciones, se evidencia la falta de un sistema de información, el problema identificado radica en el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad, tarea actualmente realizada por el Encargado de Riesgos y el Oficial de Seguridad de la Información los cuales elaboran a través de formularios manualmente, este proceder presenta muchas debilidades en cuanto a seguridad de la información debido a que contraviene la confidencialidad, integridad y disponibilidad de la información, tras el cual podría existir pérdida o duplicación de información debido a la transcripción manual y desorganizada que ocasiona retrasos en la generación de reportes planos e informes que deben ser remitidos a través del SCIP cada trimestre, al no poseer los reportes planos a través de un sistema la Entidad regulatoria demuestra que no existe capacidad de envió de información a ASFI., cada evento o incidente no registrado podría causar pérdidas económicas que afecten el estado de resultados de la Cooperativa.

1.3.1. Problema principal

En la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda., el registro de Riesgo Operativo e Incidentes de Seguridad de la Información se la realiza de forma manual, cuando acontece un evento o incidente el funcionario registra lo suscitado en un formulario o correo electrónico para entregárselo de forma física o digital al Encargado de Riesgos u Oficial de Seguridad de la Información,

acción que conlleva mucho tiempo, desorganización, acumulación y descuido. Los eventos o incidentes registrados son ordenados, revisados y solucionados a través del seguimiento que realizan el Encargado de Riesgo o el Oficial de Seguridad de la Información, también se menciona que la falta de una herramienta para el registro hace que los eventos o incidentes ni siquiera sean reportados, no existe apoyo y colaboración de parte de los demás funcionarios, por lo cual es observado por el área de Auditoría Interna de la Cooperativa. Los eventos de riesgo operativo e incidentes de seguridad de la información deberían ser reportados por los demás funcionarios, revisados por el Encargado de Riesgos u Oficial de Seguridad de la Información y solucionados por las áreas correspondientes. De esta manera el problema se basa en Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad para cumplir con lo que establece la ASFI.

1.3.2. Problemas secundarios

- Existe información dispersa lo que ocasiona desorden y retraso a la hora de su entrega.
- Gran cantidad de información que es registrada y manipulada de forma manual lo que ocasiona retrasos que impiden una solución inmediata.
- Existe descontrol de los incidentes que son realizados por un solo funcionario que interviene en la resolución lo que genera imparcialidad al momento de la toma de decisiones.
- Falta de seguimiento a los incidentes registrados lo que ocasiona desatención y pérdida de información.

De acuerdo a estos problemas se plantea la siguiente interrogante: ***¿Cómo optimizar el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.?***

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad para cumplir con lo que establece la ASFI., en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

1.4.2. Objetivos específicos

- Realizar la recopilación de información para conocer el flujo de los datos dentro de la Cooperativa y obtener puntualidad a la hora de su entrega.
- Centralizar la información de los eventos de riesgo operativo e incidentes de seguridad de la información en una Base de Datos para optimizar tiempos y dar una solución oportuna.
- Automatizar perfiles de usuario para ordenar la forma de atención que se dará a los eventos o incidentes registrados.
- Sistematizar el seguimiento de datos para controlar el estado del evento o incidente hasta darle solución.

1.5. JUSTIFICACIÓN

1.5.1. Justificación técnica

Para el funcionamiento y llegar a la misión trazada se utiliza una serie de recursos técnicos, humanos, económicos y logísticos que necesariamente asumen un conjunto de riesgos que vienen a formar parte de estos recursos, gestionarlos eficientemente y eficazmente obteniendo así, un beneficio es lo fundamental para el funcionamiento de la Cooperativa USAMA Ltda.

La entidad financiera cuenta con el equipo necesario para poner en marcha el sistema, además de contar con servidores que cuentan con licencias actualmente. Un Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad adecuada permitirá a la Unidad de Riesgos y al Área de Seguridad de la Información realizar un adecuado análisis y evaluación de los riesgos que enfrentan en la Cooperativa. El desarrollo del sistema automatizado se justifica porque cuenta con la infraestructura necesaria en cuanto a Hardware y Software para poder realizar las pruebas necesarias del sistema.

1.5.2. Justificación económica

El Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad estará desarrollado bajo Software gratuito, para optimizar los procesos de la Cooperativa evitando pérdidas económicas que afectan la imagen de la institución, amonestaciones por parte del regulador, reproceso, entre las más comunes. Beneficiará a la Cooperativa USAMA Ltda., brindando información centralizada, necesaria y actualizada reduciendo tiempos.

1.5.3. Justificación social

El Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad beneficiará a todos los funcionarios dentro de la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca Ltda., para brindar atención de calidad a los clientes pertenecientes a la Institución y así evitar que afecte a la imagen y por ende a sus representantes: Consejo de Administración y Consejo de Vigilancia, para la satisfacción y promoción ante la Asamblea General de Socios de la Cooperativa USAMA Ltda.

1.6. METODOLOGÍA

1.6.1. Metodología de desarrollo

La metodología RUP nos proporciona todas las bases para llevar al éxito la elaboración del Software, es por ello que se utilizará esta metodología al ser adaptable al contexto y necesidades de cada organización. Se centra en la producción y mantenimiento de modelos del sistema.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Es una metodología cuyo fin es entregar un producto de Software. Se estructura todos los procesos y se mide la eficiencia de la organización. Es un proceso de desarrollo de Software el cual utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

1.6.2. Métricas de Calidad

La ISO/IEC 9126, es un estándar internacional para la evolución de Software, está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. Este

estándar está pensado para los desarrolladores, adquirentes, personal que asegure la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software.

1.6.3. Costo

COCOMO II, este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permitirá estimar el esfuerzo y tiempo requerido para desarrollar el producto.

1.6.4. Seguridad

Se implementará controles de acceso en el software lo cual ayudará a proteger la información manteniendo su integridad para evitar los accesos no autorizados. También se creará roles para controlar a través de la función o rol del usuario que requiere dicho acceso.

1.6.5. Pruebas de Software

Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto.

1.6.5.1. Caja negra.

En pruebas de software, conociendo una función específica para la que fue diseñado el producto, se pueden diseñar pruebas que demuestren que dicha función está bien realizada. Dichas pruebas son llevadas a cabo sobre la interfaz del software, es decir, de la función, actuando sobre ella como una caja negra, proporcionando unas entradas y estudiando las salidas para ver si concuerdan con las esperadas.

1.6.5.2. Caja blanca.

Se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. El proceso de pruebas en el ciclo de vida aconseja que, durante la ejecución de procesos principales de la organización, utilizar los procesos de soporte, entre ellos las pruebas de validación y verificación.

1.6.6. Método de ingeniería

1.6.6.1. Identificación del problema.

La Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. necesita un método que gestione la información de sus eventos de riesgo y sus incidentes de seguridad de la información de manera precisa y segura. De acuerdo a esta problemática se plantea la siguiente interrogante:

¿Cómo optimizar el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.?

1.6.6.2. Recopilación de la información necesaria.

El proceso de recolección o recopilación de información fue la aplicación de una investigación dirigida a los funcionarios de la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. y a la Recopilación de Normas de Servicios Financieros con el fin de obtener y conocer cómo se maneja el proceso de Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad, además de su envío y validación.

1.6.6.3. Búsqueda de soluciones creativas.

Generalmente promueve un proceso de administración de proyectos disciplinados que da pie a la frecuente inspección y adaptación, una filosofía de liderazgo que promueve el trabajo en equipo, una serie de buenas prácticas que permiten la rápida entrega de un sistema de alta calidad, este equipo será conformado por el desarrollador, el jefe de Sistemas, el Encargado de Riesgos, el Oficial de Seguridad de la Información, la Auditora Interna y los tutores. La siguiente problemática es: ¿Cómo optimizar el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.?

1.6.6.4. Evaluación y elección de la solución.

Al realizar un análisis exhaustivo de la problemática existente se podrá obtener los requerimientos y se llevará a cabo la mejor solución con la finalidad de que ésta cumpla las expectativas planteadas.

1.6.6.5. Preparación de reportes, planos y especificaciones.

Para la presentación de los distintos reportes, planos y especificaciones, se las deberá realizar al momento en el que se empiece con el proyecto.

1.6.6.6. Implementación del proyecto.

La metodología RUP presenta las características que permitirán la implementación de los requerimientos en forma efectiva y a dar solución a las observaciones ASFI. existentes, se estará realizando la implementación el cual permitirá automatizar y mostrar los eventos de riesgo e incidentes de seguridad de la información para su envió a través del SCIP a la ASFI. trimestral y anualmente.

Al concluir con esos pasos, se procederá a la implementación del sistema, con el objetivo de determinar si el proyecto de grado cumplió o no con las especificaciones consideradas al inicio del desarrollo de la presente.

1.7. HERRAMIENTAS

Las herramientas que se utilizarán son:

- ✓ Sistema Operativo: Windows 10
- ✓ Base de Datos: MySQL.

  MYSQL, es un motor de base de datos con el cual se trabajará.
- ✓ Lenguajes: PHP, HTML, JAVASCRIPT, CSS.

  PHP, es un lenguaje de programación que nos permitirá realizar el sistema base.

  HTML, es un lenguaje de marcado de hipertexto ayudara a php a mostrarse en modo web.

  JAVASCRIPT, es un lenguaje de programación que se utilizará como complemento de HTML Y CSS.

  CSS, nos ayudará en el diseño dinámico del sistema.
- ✓ Framework: Laravel y Boostrap

1.8. LÍMITES Y ALCANCES

1.8.1. Límites

El Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad:

- El sistema no estará enlazado al sistema de la Entidad AFCOOP[9].
- El sistema no estará enlazado al sistema de la Entidad FECAC[10].
- Tampoco se encargará del riesgo de crédito, riesgo de mercado o riesgo de liquidez.
- No se desarrollará un sistema de información histórica de eventos de riesgos pasados.

1.8.2. Alcances

Con el funcionamiento del Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad:

- Se administrarán los eventos de riesgos operativo y los incidentes de seguridad de la información por módulos: (funcionarios - Evento de Riesgo Operativo, Incidente de Seguridad de la información), (Encargado de Riesgos – Evento de Riesgo Operativo), (Oficial de Seguridad de la Información – Incidente de Seguridad de la Información) y (Administrador – Evento de Riesgo Operativo, Incidente de Seguridad de la Información, Usuarios, Backups).
- Se obtendrá a tiempo acciones preventivas para evitar pérdidas que afecten el estado de ganancias y pérdidas en la Entidad.
- Se podrá demostrar que existe capacidad de envió de información trimestral a la entidad regulatoria.
- Con el Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad se espera evitar las observaciones de la ASFI.

1.9. APORTES

Sin duda, el aporte de un sistema permitirá que los funcionarios de la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda., independientemente de su cargo, tengan a la mano una herramienta necesaria para evitar que la Entidad incurra en pérdidas por fraude interno o externo, fallas en los procesos, sistemas, que no se presenten eventos internos, externos. El proyecto

[9] AFCOOP. Autoridad de Fiscalización y Control de Cooperativas.
[10] FECAC. Federación de Cooperativas de Ahorro y Crédito de La Paz.

propuesto ayudará a minimizar tiempos, reducir la cantidad de papeleo acumulado, optimizará el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad, creará acciones preventivas para evitar pérdidas en la Entidad y ayudará a la organización en su envió trimestral y anual.

CAPÍTULO II

CAPÍTULO II
MARCO TEÓRICO

En este capítulo se muestran los fundamentos teóricos que permitirán dar forma al Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

2.1. ASPECTOS CONCEPTUALES

2.1.1. Sistema

Es un conjunto organizado de cosas o partes interactuantes e interdependientes, que se relacionan formando un todo unitario y complejo. Cabe aclarar que las cosas o partes que componen al sistema, no se refieren al campo físico (objetos), sino más bien al funcional. De este modo las cosas o partes pasan a ser funciones básicas realizadas por el sistema. Podemos enumerarlas en: entradas, procesos y salidas (Bertalanffy, 2010).

Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. Siguiendo esta propuesta, podemos decir que un sistema es la organización de partes interactuantes e interdependientes que se encuentran unidas y relacionadas para formar una célula compleja. Con esto nos referimos a un grupo de elementos que realizan actividades para alcanzar un objetivo común, ya sea operando sobre los datos, la energía o la materia para suministrar información (Márquez, 2015).

De acuerdo con (Bertalanffy, 2010) el sistema es un conjunto ordenado de piezas que interactúan para moldear un todo, el cual posee una entrada, un proceso y una salida. Por otro lado, según (Marqués, 2015) un sistema es un conjunto de elementos que se interrelacionan para lograr un objetivo, un sistema es una estructura de piezas que combinadas forman una célula compleja.

2.1.2. Información

El concepto de información, tanto desde el punto de vista popular como del científico, involucra un proceso de reducción de incertidumbre. En el lenguaje diario, la idea de información se encuentra relacionada a la de novedad y utilidad, pues

información es el conocimiento (no cualquier conocimiento) disponible para el uso inmediato y que permite orientar la acción, al reducir el margen de incertidumbre que cerca las decisiones cotidianas. En la sociedad moderna, la importancia de la disponibilidad de la información amplia y variada crece proporcionalmente al aumento de la complejidad de la propia sociedad (Chiavenato, 2007).

La información es un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo (Thompson, 2008).

En ese sentido según (Chiavenato, 2007) la información incluye el proceso de disminución de la duda, información se asocia con innovación, porque la información es conocimiento. De acuerdo con (Thompson, 2008) información es la agrupación de datos sobre un evento, que posee un significado, que puede estar encaminado a reducir dudas o aumentar el conocimiento.

2.1.3. Sistema de Información

Podemos plantear la definición técnica de un sistema de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos. Los sistemas de información contienen información sobre personas, lugares y cosas importantes dentro de la organización, o en el entorno que la rodea (Laudon & Laudon, 2004).

Sistemas de información, al comprender las palabras “información” y “sistema”, la definición de un sistema de información es casi intuitiva: un sistema de información (IS) está formado por todos los componentes que colaboran para procesar los datos y producir información. Casi todos los sistemas de información empresariales están

integrados por muchos subsistemas con metas secundarias, todas las cuales contribuyen a la meta principal de la organización (Oz, 2008).

Por lo tanto (Laudon & Laudon, 2004) menciona que un sistema de información es la agrupación de datos que recolecta, procesa, almacena y distribuye información con el fin de ayudar en las decisiones a tomar. En ese sentido (Oz, 2008) nos indica que la definición de un sistema de información es casi intuitiva: contar con todos los elementos para procesar datos y generar información.

2.1.4. Ingeniería de Software

La Ingeniería de/l Software es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la Zngenierh de/l Software como una nueva área de la ingeniería, y el ingeniero de/l software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidades que cumplir, junto a una, ya, reconocida consideración social en el mundo empresarial y, por suerte, para esas personas con brillante futuro.

La Ingeniería de/l Software trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc.(Pressman, 2005).

La ingeniería de software es una tecnología con varias capas. Como se aprecia en la figura 1.3, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares10 alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.

Figura 2

Capas de la Ingeniería de Software

Nota. Capas de la Ingeniería de Software (Figura 1.3. del Libro de Pressman,2010)

El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base
para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas. Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software (Pressman, 2010).

De acuerdo con (Pressman,2005) la Ingeniería de Software es una disciplina que implica el uso de estructuras, herramientas y técnicas para construir programas informáticos. Así mismo, incluye el análisis previo de la situación, la redacción del proyecto, la creación del software y las pruebas necesarias para garantizar el correcto funcionamiento del software antes de poner el sistema en funcionamiento. Por otro lado, según (Pressman, 2010) La Ingeniería de Software es la ingeniería que estudia todo lo relacionado con la informática o sistemas de computación, con una orientación metódica, ordenada y cuantificable al incremento, ejecución y conservación del Software.

2.1.5. Seguimiento

El seguimiento consiste en el análisis y recopilación sistemáticos de información a medida que avanza un proyecto. Su objetivo es mejorar la eficacia y efectividad de un proyecto y organización. Se basa en metas establecidas y actividades planificadas durante las distintas fases del trabajo de planificación. Ayuda a que se siga una línea de trabajo, y además, permite a la administración conocer cuando algo no está funcionando. Si se lleva a cabo adecuadamente, es una herramienta de incalculable valor para una buena administración y proporciona la base para la evaluación. Te permite determinar si los recursos disponibles son suficientes y están bien administrados, si tu capacidad de trabajo es suficiente y adecuada, y si estás haciendo lo que habías planificado (CIVICUS, 2007).

El seguimiento es una acción permanente a lo largo del proceso de los proyectos, permite una revisión periódica del trabajo, tanto en su eficiencia en el manejo de recursos humanos y materiales, como de su eficacia en el cumplimiento de los objetivos propuestos. Es de vital importancia que el seguimiento se realice como una parte integrante del proceso del proyecto, acordada con los responsables de la gestión, para que no suceda como una mera supervisión. Recordemos que la función del sistema consiste en aportar aprendizaje institucional y no en emitir dictámenes. Los propósitos del seguimiento son: Fomentar la cultura de la evaluación, la gestión del desempeño y la rendición de cuentas en función de los resultados esperados. Alinear la evaluación con el ciclo de los proyectos, como un elemento sustantivo de la

planificación estratégica. Alentar el aprendizaje institucional de todos los actores involucrados en el proyecto con base en las evaluaciones efectivas y de calidad. Promover el uso de la evidencia proporcionada por el seguimiento. Elegir los resultados pertinentes y demostrar cómo y por qué producen los resultados previstos o cómo mejoran lo esperado (Van de Velde, 2009).

De acuerdo con (CIVICUS, 2007) el seguimiento es el estudio ordenado de la recopilación de información durante el proyecto, su finalidad es aumentar la eficacia y la eficiencia de los proyectos, si se hace correctamente es una herramienta valiosa para la buena gobernanza. Aprueba los recursos libres, si tu trabajo es adecuado y suficiente y si estás siguiendo tu plan. Por otro lado, según (Van de Velde, 2009) el seguimiento es una actividad continua que permite revisiones periódicas del trabajo, es fundamental que el seguimiento sea parte integral del proceso del proyecto y este en constante relación con los responsables de la gestión.

2.1.6. Control

Se entiende por control el hecho de procurar, mediante un algoritmo (de control), que una determinada variable del proceso (variable controlada) se mantenga igual (o dentro de unos márgenes fijados) a un valor (punto de consigna) modificando si es necesario una variable de entrada (variable manipulada). Algunas de las ventajas del control automático son las siguientes: Aumento en la cantidad o número de productos. Mejora en la calidad de productos. Economía de materiales. Economía de energía o potencia. Economía de equipos industriales. Reducción en la inversión de mano de obra en tareas no especializadas. Facilitar el trabajo del obrero (Coronel Delgado & Peralta Espinoza, 2009).

El control es una etapa primordial en la administración, pues, aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, el ejecutivo no podrá verificar cuál es la situación real de la organización y no existe un mecanismo que se cerciore e informe si los hechos van de acuerdo con los objetivos. El concepto de control es muy general y puede ser utilizado en el contexto organizacional para evaluar el desempeño general frente a un plan estratégico. La palabra control tiene muchas connotaciones y su significado depende

de la función o del área en que se aplique; puede ser entendida: Como la función administrativa que hace parte del proceso administrativo junto con la planeación, organización y dirección, y lo que la precede (Yuri Cabrera, 2016).

En ese sentido según (Coronel Delgado & Peralta Espinoza, 2009) el Control es intentar mantener igual un proceso. De acuerdo con (Yuri Cabrera, 2016) el control es una fase esencial en la administración, porque por más que una empresa tenga todo bien organizado no funcionara si no se encuentra ningún mecanismo para determinar si los hechos están de acuerdo con las metas.

2.1.7. Seguimiento y Control

El seguimiento consiste básicamente en el análisis de la información, mediante proyecciones relacionadas con los tiempos que son generadas en un proyecto, para la identificación temprana de riesgos y desviaciones respecto al plan. Por su parte el control comprende el desarrollo de las actuaciones para conseguir que lo planificado y esperado ocurra. Por lo tanto, controlar no significa sólo identificar, sino que la esencia del control supone indagar en las causas que puedan conllevar a definir las acciones e implementarlas de manera que sus efectos lleven a minimizar riesgos o peligros dentro de un proyecto establecido (Sanz, 2016).

De acuerdo con (Sanz, 2016) el seguimiento radica en examinar la información para identificar los riesgos y lejanía del plan. Por otro lado, el control en sí implica la creación de actividades para garantizar que suceda lo que se planea y se espera. Así, el control no significa simplemente identificación, sino que la esencia del control incluye el estudio de las causas que pueden conducir a las acciones identificadas y su implementación de tal forma que sus efectos generen la minimización de riesgos.

2.1.8. Riesgo

La concepción del riesgo es una abstracción de origen completamente humano. El concepto de riesgo, asociado con la idea de porvenir sin certeza, ha estado presente desde siempre en las sociedades humanas. Su propia concepción implica un devenir de los acontecimientos. En esta característica temporal radica la variabilidad que dificulta su predicción. Así mismo, un acontecimiento que produce una consecuencia

no deseada sobre el Hombre está asociado al espacio físico donde éste desarrolla sus actividades. La distribución espacial de esta afectación sobre el territorio geográfico también se caracteriza por su gran variabilidad. Estas dos características, temporal y espacial, del riesgo lo convierten en un concepto esencialmente dinámico (Soldano, 2009).

La palabra riesgo es tan antigua como la propia existencia humana. Podemos decir que con ella se describe, desde el sentido común, la posibilidad de perder algo (o alguien) o de tener un resultado no deseado, negativo o peligroso. El riesgo de una actividad puede tener dos componentes: la posibilidad o probabilidad de que un resultado negativo ocurra y el tamaño de ese resultado. Por lo tanto, mientras mayor sea la probabilidad y la pérdida potencial, mayor será el riesgo. Cada vez que tomamos una decisión y valoramos la relación costos-beneficios, no estamos sino evaluando los riesgos que corremos con esa decisión y las ventajas o desventajas que esta nos puede traer. Es decir, funcionamos cotidianamente con la noción de riesgos, aunque no seamos conscientes de ello en todo momento. Por lo tanto, ni la palabra riesgo ni el fenómeno que se describe con ella son nuevos para nuestro entendimiento, al contrario, el ser humano desde sus inicios como especie convivía naturalmente con los riesgos y reaccionaba intuitivamente ante ellos (Echemendía Tocabens, 2011).

Por lo tanto (Soldano, 2009) menciona que riesgo se encuentra relacionado con la idea de un futuro incierto. Su concepto sugiere el futuro de los acontecimientos. En ese sentido (Echemendía Tocabens, 2011) nos indica que riesgo describe la probabilidad de corromperse algo u obtener un resultado indeseable. El riesgo es la probabilidad de un resultado negativo.

2.1.9. Operativo

La palabra operativo comenzó siendo un adjetivo cuyo significado, según el DRAE es “preparado o listo para ser utilizado o entrar en acción” y poco a poco fue convirtiéndose en sustantivo y usado como tal en el ámbito militar y policial con el significado de ‘conjunto de acciones coordinadas para conseguir un fin. Preparado o listo para ser utilizado o entrar en acción (Asociacion de Academias de la Lengua Española, 2021).

De acuerdo con la (Asociación de Academias de la Lengua Española, 2021) operativo significaba "listo para usar o para comenzar a operar", gradualmente se convirtió en un sustantivo. La palabra operativo se la define como listo para usar o para empezar a trabajar.

2.1.10. Riesgo Operativo

Riesgo operativo se define como el riesgo de pérdida debido a las deficiencias o a fallas de los procesos, el personal y los sistemas internos, o bien a causa de acontecimientos externos. El tipo y frecuencia de eventos que abarca es muy diverso. Esta definición incluye el riesgo legal, pero excluye el estratégico y el de reputación. Del riesgo operativo se pueden destacar las siguientes características: el riesgo operativo es el más antiguo de todos y está presente en cualquier clase de negocio y casi en toda actividad; es inherente a toda actividad en que intervengan personas, procesos y plataformas tecnológicas; es complejo, como consecuencia de la gran diversidad de causas que lo originan; y las grandes pérdidas que ha ocasionado a la industria financiera muestran el desconocimiento que de él se tiene y la falta de herramientas para gestionarlo. Es conveniente indicar en este punto la principal diferencia relevante para el modelado del riesgo (Nuñez Mora & Chavez Gudiño, 2010).

El riesgo operacional se define como “el riesgo de sufrir pérdidas debido a la inadecuación o a fallos de los procesos, personas o sistemas internos o bien a causa de acontecimientos externos”. El riesgo operacional se presenta a través de diferentes eventos que pueden surgir por fallas, errores, omisiones, uso no autorizado o fraude, de las personas, procesos, sistemas o externos, afectando una o más líneas de negocio generando uno o más efectos que reducen valor a la organización (Velezmoro La Torre, 2010).

En ese sentido según (Núñez Mora & Chávez Gudiño, 2010) riesgo operacional significa riesgo de pérdida como resultado de debilidades en los procesos internos, personas y sistemas. La regularidad de eventos que concibe es amplia siendo el riesgo operacional el más antiguo. De acuerdo con (Vélezmoro La Torre, 2010) riesgo operacional se la describe como “el riesgo de pérdida derivado de deficiencias o fallas

en procesos, personas o sistemas". El riesgo operacional se manifiesta por diversos sucesos que pueden ocurrir por fallas que reducen el valor de un entorno.

2.1.11. Incidente

Se denomina incidente "cualquier suceso no esperado ni deseado que NO dando lugar a pérdidas de la salud o lesiones a las personas puede ocasionar daños a la propiedad, equipos, productos o al medio ambiente, pérdidas de producción o aumento de las responsabilidades legales" (Fernández et al., 2008).

Acontecimiento o anomalía que se produce de forma imprevista durante la realización del trabajo, que no produce daños al trabajador aunque tiene capacidad de producirlos en circunstancias ligeramente diferentes y puede ser considerada fuente de riesgo para los trabajadores (Cortés, 2017).

Acontecimiento no deseado que resulta o puede resultar en pérdidas. Si el resultado se presenta como lesiones a personas y/o daños a la propiedad (pérdidas) se denomina accidente (Mutual de Seguridad, 2015).

Por lo tanto (Fernández et al., 2008) menciona que incidente se define como "cualquier evento inesperado que no resulte en pérdida de la salud y que pueda generar daños a la propiedad, equipo, producto o medio ambiente. En ese sentido (Cortés, 2017) nos indica que incidente es un evento o situación inusual que ocurre inesperadamente durante la ejecución de un trabajo. De acuerdo con (Mutual de Seguridad, 2015) incidente es un suceso inesperado que causa o puede causar pérdida.

2.1.12. Seguridad

La seguridad tiene un significado muy amplio, y el clásico, de acuerdo con la teoría liberal, es que constituye la esencia y el deber ser del Estado. Esta visión se dividió en dos áreas: la defensa frente a amenazas externas al Estado –representadas principalmente por otros Estados–, que es materia de la seguridad nacional, y la seguridad interior, que es responsabilidad del gobierno y forma parte de la seguridad pública. Sin embargo, en un contexto histórico en el que las amenazas a la seguridad ya no son producto de la lucha entre Estados, donde no existe una separación clara,

sino una relación cada vez más fuerte entre asuntos internos y externos o locales y globales, y donde los nuevos riesgos provienen de actores que buscan permanecer ocultos, han surgido nuevas propuestas para el término identificadas, sobre todo, en la seguridad interior, la humana y la democrática (Montero Bagatella, 2013).

El concepto de seguridad es discutido en Ciencias Sociales llegando incluso a veces a resultar equivoco. Uno de los problemas principales al abordar la seguridad consiste en determinar cuál es el objeto referente de la misma. Para este estudio comparativo se entenderá seguridad como: «la capacidad de las personas, los Estados o las sociedades de librarse de las amenazas y de mantener su independencia en lo que se refiere a su identidad y a su integración funcional frente a fuerzas de cambio consideradas hostiles. Securitización es un proceso por el que se da calidad o estatus de asunto de seguridad a una cuestión que puede atentar contra la supervivencia de un ente, ya sea: individuos, una colectividad, Estados o la humanidad (Penalva Lucas, 2017).

De acuerdo con (Montero Bagatella, 2013) el significado clásico de seguridad es el deber del estado. Por otro lado, según (Penalva Lucas, 2017) la definición de seguridad es objeto de debate. Uno de los mayores problemas con la seguridad es definir lo que cubre. Seguridad es el estado en el cual los peligros son controlados.

2.1.13. Incidente de Seguridad

Un incidente de seguridad de la información se define como un acceso, intento de acceso, uso, divulgación, modificación o destrucción no autorizada de información; un impedimento en la operación normal de las redes, sistemas o recursos informáticos; o una violación a una Política de Seguridad de la Información de la entidad. La notificación de los incidentes permite responder a los mismos en forma sistemática, minimizar su ocurrencia, facilitar una recuperación rápida y eficiente de las actividades minimizando la pérdida de información y la interrupción de los servicios, y el proceso de tratamiento de incidentes, y manejar correctamente los aspectos legales que pudieran surgir durante este proceso (Seguridad y Privacidad de la Informacion, 2009).

Evento o una serie de eventos de seguridad de la información no deseados o inesperados, que tienen una probabilidad significativa de comprometer las operaciones del negocio y amenazar la seguridad de la información (Consejo para las Tecnologias de Informacion y Comunicacion, 2010).

En ese sentido según la (Seguridad y Privacidad de la Información, 2009) incidente de seguridad de la información se entiende como el acceso no autorizado, la interrupción al funcionamiento adecuado de una red, sistema o recurso informático. De acuerdo con el (Consejo para las Tecnologías y Comunicación, 2010) un incidente de seguridad de la información es algo no deseado, algo no inesperado, es una serie de eventos con un alto potencial de poner en peligro la seguridad de la información.

2.1.14. Cooperativa

Una cooperativa es una asociación autónoma de personas que se reúnen de forma voluntaria para satisfacer sus aspiraciones económicas, sociales y culturales, mediante una organización de propiedad conjunta y de gestión democrática sin fines de lucro. Las cooperativas representan un modelo asociativo en el que los objetivos económicos y empresariales se integran con otros de carácter social, consiguiendo de esta forma un crecimiento basado en el empleo, la equidad y la igualdad. Otras definiciones sobre cooperativas refieren a una asociación sin fines de lucro en la cual los trabajadores o usuarios, según el caso, son simultáneamente aportantes y gestores de la empresa. Es creada con el objeto de producir bienes o servicios para satisfacer las necesidades de sus asociados y de la comunidad en general (INEI Instituto Nacional de Estadistica e Informatica, 2010).

La cooperativa es una sociedad de capital variable, con estructura y gestión democrática, constituida por personas físicas o jurídicas, para prestar servicios y satisfacer sus necesidades y aspiraciones económicas y sociales, y en interés por la comunidad, mediante una empresa conjunta (Castilla-La Mancha, 2021).

Por lo tanto, la| (INEI Instituto Nacional de Estadística e Informática, 2010) menciona que Cooperativa es una agrupación de personas se unen voluntariamente para perseguir sus aspiraciones económicas, sociales y culturales a través de una

organización. En ese sentido (Castilla-La Mancha, 2021) nos indica que una cooperativa es una empresa de capital variable con organización democrática.

2.1.15. Cooperativa de Ahorro y Crédito

Las cooperativas de ahorro y crédito o, simplemente, cooperativas de crédito son sociedades cooperativas cuyo objeto social es servir las necesidades financieras de sus socios y de terceros mediante el ejercicio de las actividades propias de las entidades de crédito (INEC, 2017).

Se denominan Cooperativas de Ahorro y Crédito Abiertas cuando sus operaciones financieras de Ahorro y Crédito son realizadas con los socios, público en general sin discriminación de ninguna clase y con diversas entidades financieras, nacionales o extranjeras para su funcionamiento. Están obligadas a agregar en su denominación la palabra “Limitada” o la abreviatura “Ltda.” adoptando el régimen de responsabilidad Limitada (Aramayo F., 2006).

De acuerdo con (INEC, 2017) las cooperativas de ahorro y crédito son entidades cuyo propósito es satisfacer las necesidades financieras de sus socios. Por otro lado, según (Aramayo F., 2006) cooperativa de ahorro y crédito es para socios sin discriminación alguna, se denominan Ltda. por ser un sistema de responsabilidad limitada.

2.2. CICLO DE VIDA DE UN SISTEMA

Los proyectos software, tiene como finalidad obtener un producto, proceso que es necesario generar a través de diferentes actividades. Alguna de estas actividades puede agruparse en fases. Al conjunto de las fases empleadas se conoce como “ciclo de vida”. Periodo de tiempo que comienza con la decisión de desarrollar el producto software y termina cuando el software es entregado. Esta metodología es un enfoque por fases, que puede usar diferentes métodos (cascada, lineal, en V y otros).

Periodo de tiempo que comienza cuando el producto software es concebido y termina cuando el software no está disponible permanentemente para el usuario (retirada del software). El desarrollo de Software va unido a un “ciclo de vida”, compuesta por una serie de fases (etapas) que comprenden todas las actividades,

desde el momento en que surge la idea de crear un nuevo producto, hasta aquel en que el producto deja definitivamente de ser utilizado (Cotaña, 2014).

El software no se estropea. La gráfica de fallos en función del tiempo, tendría forma de caída desde el principio, hasta mantenerse estable por tiempo casi indefinido. El software no es susceptible a los males del entorno que provocan el deterioro del hardware. Los efectos no detectados harán que falle el programa durante las primeras etapas de su vida, sin embargo, una vez corregidas, no se producen nuevos errores.

Aunque no se estropea, si puede deteriorarse. Esto sucede debido a los cambios que se efectúan durante su vida. Cuando un componente hardware se estropea, se cambia por otro que actúa como una "pieza de repuesto", mientras que, para el software, no es habitual este proceso, lo cual significa que el mantenimiento de los programas es muy complejo. La mayoría del software es a medida (Anonimus, 1955).

En ese sentido según (Cotaña, 2014) el objetivo de un proyecto de software es lograr un producto. Ciclo de vida es un período de tiempo para el desarrollo de productos de software y la finalización al decidir entregar software. El tiempo entre el desarrollo de un producto de software y el final cuando el software deja de estar disponible para los usuarios de forma permanente (fin del software). De acuerdo con (Anonymus, 1955) el software no falla, el software está menos expuesto a los factores ambientales que provocan la degradación del hardware. Ciclo de vida del desarrollo de software es la estructura que contiene los procesos, actividades y tareas relacionadas con el desarrollo y mantenimiento de un producto de software.

2.2.1. Ciclo de vida incremental

El modelo incremental combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento de software. Por ejemplo, el software de tratamiento del texto desarrollado con el paradigma incremental podría extraer funciones de gestión de archivos básicos y de producción de documentos en el primer incremento; funciones de edición más sofisticadas y de

producción de documentos en el segundo incremento; corrección ortográfica y gramatical en el tercero; y una función avanzada de esquema de página en el cuarto. Se debería tener en cuenta que el flujo del proceso de cualquier incremento puede incorporar el paradigma de construcción de prototipos. “el modelo incremental entrega software en partes pequeñas, pero utilizables, llamadas incrementos. En general cada incremento se constituye sobre aquél que ya ha sido entregado (Canepa Sáenz & García González, 2011).

Este modelo de ciclo de vida se basa en la filosofía de construir incrementando las funcionalidades del programa. Se realiza construyendo por módulos que cumplen las diferentes funciones del sistema. Esto permite ir aumentando gradualmente las capacidades del software. Este ciclo de vida facilita la tarea del desarrollo permitiendo a cada miembro del equipo desarrollar un módulo particular en el caso de que el proyecto sea realizado por un equipo de programadores. Es una repetición del ciclo de vida en cascada, aplicándose este ciclo en cada funcionalidad del programa a construir. Al final de cada ciclo le entregamos una versión al cliente que contiene una nueva funcionalidad. Este ciclo de vida nos permite realizar una entrega al cliente antes de terminar el proyecto.

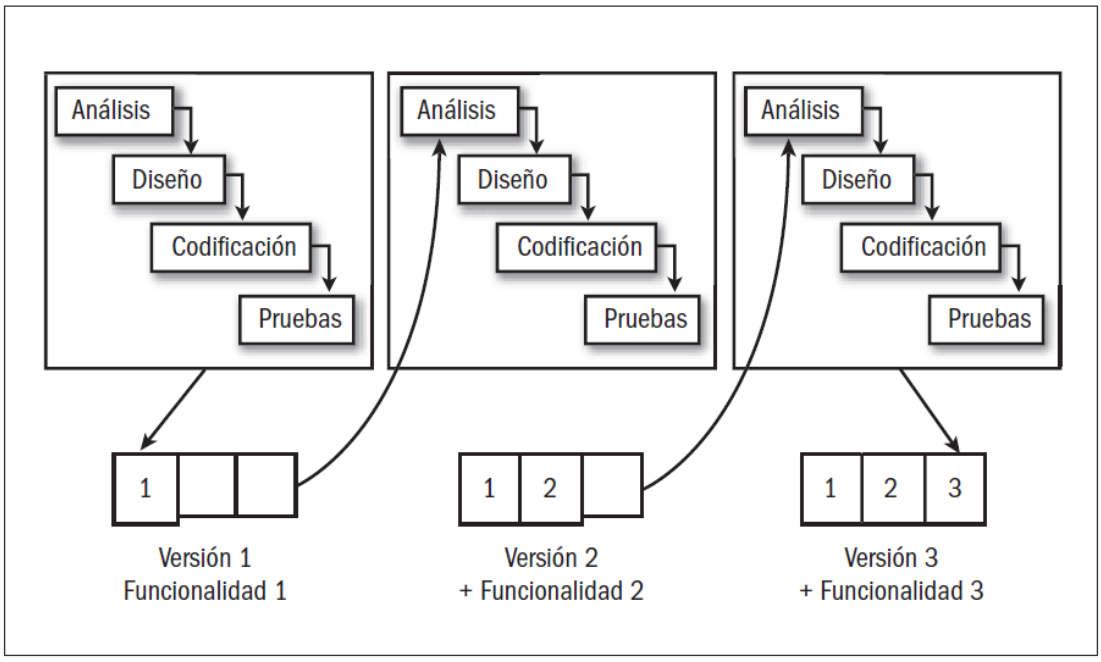
El modelo de ciclo de vida incremental nos genera algunos beneficios tales como los que se describen a continuación:

- Construir un sistema pequeño siempre es menos riesgoso que construir un sistema grande.
- Como desarrollamos independientemente las funcionalidades, es más fácil relevar los requerimientos del usuario.
- Si se detecta un error grave, sólo desechamos la última iteración.

Este modelo de ciclo de vida no está pensado para cierto tipo de aplicaciones, sino que está orientado a cierto tipo de usuario o cliente. Podremos utilizar este modelo de ciclo de vida para casi cualquier proyecto, pero será verdaderamente útil cuando el usuario necesite entregas rápidas, aunque sean parciales (Cantone, 2006).

Figura 3

Modelo Ciclo de vida incremental



Nota. Modelo Ciclo de vida incremental (Cantone, 2006)

En ese sentido según (Canepa Sáenz & García González, 2011) el ciclo de vida Incremental es un proceso bajo el cual se produce una aplicación parcial y poco a poco se aumenta la funcionalidad o el rendimiento planificación con anterioridad mediante incrementos subsecuentes. De acuerdo con (Cantone, 2006) el ciclo de vida Incremental es aquel que repite las actividades del proyecto en fases o iteraciones y en cada una de ellas se aumenta el entendimiento del producto por parte del equipo del proyecto. Las iteraciones desarrollan el producto a través de una serie de ciclos repetidos que van añadiendo sucesivamente funcionalidad al producto.

2.3. METODOLOGÍA

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto

de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea (Maida & Pacienzia, 2015).

2.3.1. Metodologías de Desarrollo

Las metodologías para el desarrollo de sistemas ayudan a la construcción de software con un importante aporte de calidad, señalando los diferentes pasos y actividades a realizar hasta lograr el producto buscado. Tal producto tiene origen de la necesidad de los usuarios para soportar procesos importantes de las actividades que estos últimos realizan, como ser: almacenamiento masivo de información, seguimiento y control de los diferentes procesos automatizados, cálculos periódicos y precisos, emisión de reportes actuales e históricos, etc.

Por lo general estas metodologías solo contemplan los términos técnicos de la creación de software, abarcando así las etapas conocidas como análisis, diseño, implementación e implantación, desde diferentes enfoques (MDS-ALFA, 2009).

Son muchas las definiciones que se pueden encontrar sobre qué es una metodología de desarrollo de software, tan es así que hay quienes hablan de ella como organización sistemática para el ciclo de vida del sistema y sus partes, pero también hay quienes la definen como conjunto de herramientas, procedimientos y técnicas para lograr el desarrollo de un nuevo software. Sin embargo, para hacerlo sencillo, se puede definir la metodología de software como un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la Ingeniería de Software.

De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada.

-Cuáles son los objetivos de una Metodología de Desarrollo de Software.

- Establecer acertadamente cada uno de los requisitos de un sistema software.
- Suministrar un método sistemático de desarrollo de manera tal que se pueda controlar su proceso.

- Hacer la construcción de un sistema de software dentro de un tiempo apropiado y costos aceptables.
- Hacer la construcción de un sistema que además de estar bien documentado, sea fácil de mantener.
- Ayudar a identificar, lo antes posible, cualquier cambio que sea necesario realizar dentro del proceso de desarrollo.
- Proveer un sistema que satisfaga a las personas afectadas por el mismo.

-Qué aspectos debería cubrir una Metodología de Desarrollo de Software.

- Un proceso de ciclo de vida completo, es decir, uno que comprenda aspectos tanto del negocio como aquellos que son de origen técnico.
- Un conjunto completo de conceptos y modelos que sean internamente consistente.
- Una descripción completa de artefactos a desarrollar Un conjunto de técnicas probadas.
- Identificación de los roles organizacionales
- Guías para la gestión de proyectos y aseguramiento de la calidad Asesoramiento para la gestión de bibliotecas y reutilización (Mega Practical, 2010).

En ese sentido según (MDS-ALFA, 2009) las metodologías de desarrollo de sistemas ayudan a crear software para lograr el resultado deseado. Todo esto surge de importantes procesos como: almacenamiento de información, seguimiento de procesos, cálculos periódicos, publicación de informes, etc. Las metodologías abarcan etapas de análisis, diseño, implementación e implantación. De acuerdo con (Mega Practical, 2010) existen muchas definiciones sobre la metodología de desarrollo de software, como un conjunto de herramientas, procedimientos y técnicas. La metodología del software puede definirse como un método, una forma de interpretar la realidad, o una disciplina afín.

2.3.2. Metodología RUP

RUP es un proceso de Ingeniería de Software que proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de un desarrollo

organizado. Su objetivo es asegurar la producción de Software de alta calidad que cumpla las necesidades de los usuarios finales, dentro de unos tiempos y presupuestos predecibles.

RUP promueve la productividad del trabajo en equipo proporcionando a cada miembro del equipo un fácil acceso a una base de conocimiento con una serie de directrices, plantillas y herramientas para actividades de desarrollo críticas. No importa si los miembros del equipo trabajan en distintas disciplinas de un proyecto, como requisitos, diseño o pruebas, los distintos miembros del equipo comparten un lenguaje común, procedimientos y punto de vista sobre cómo desarrollar el Software.

Las actividades crean y mantienen modelos, que son representaciones, semánticamente ricas, de un sistema Software en desarrollo. RUP no se centra en la producción de una gran cantidad de documentos, enfatiza el desarrollo y mantenimientos de los modelos.

RUP es soportado por herramientas que automatizan grandes partes del proceso. Son usadas para crear y mantener artefactos, modelos en particular, del proceso de Ingeniería de Software: modelado visual, programación, testing, etc. Tienen un gran valor en el apoyo a la contabilidad asociada a la gestión de cambios, así como la gestión de la configuración que acompaña a cada iteración. Es también un proceso configurable. No todos los procesos son adaptables para cualquier desarrollo de Software, sin embargo, RUP es válido tanto para pequeños equipos como en grandes organizaciones de desarrollo (López Rosciano & Pech Montejo, 2015).

El Proceso Unificado es un proceso de desarrollo de Software: “conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema Software”. RUP es un marco genérico que puede especializarse para una variedad de tipos de sistemas, diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y diferentes tamaños de proyectos. RUP está basado en componentes. El Software está formado por componentes Software interconectados a través de interfaces. RUP está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema.

Fases

Cada ciclo constas de cuatro fases: inicio, elaboración, construcción, y transición.

Figura 4

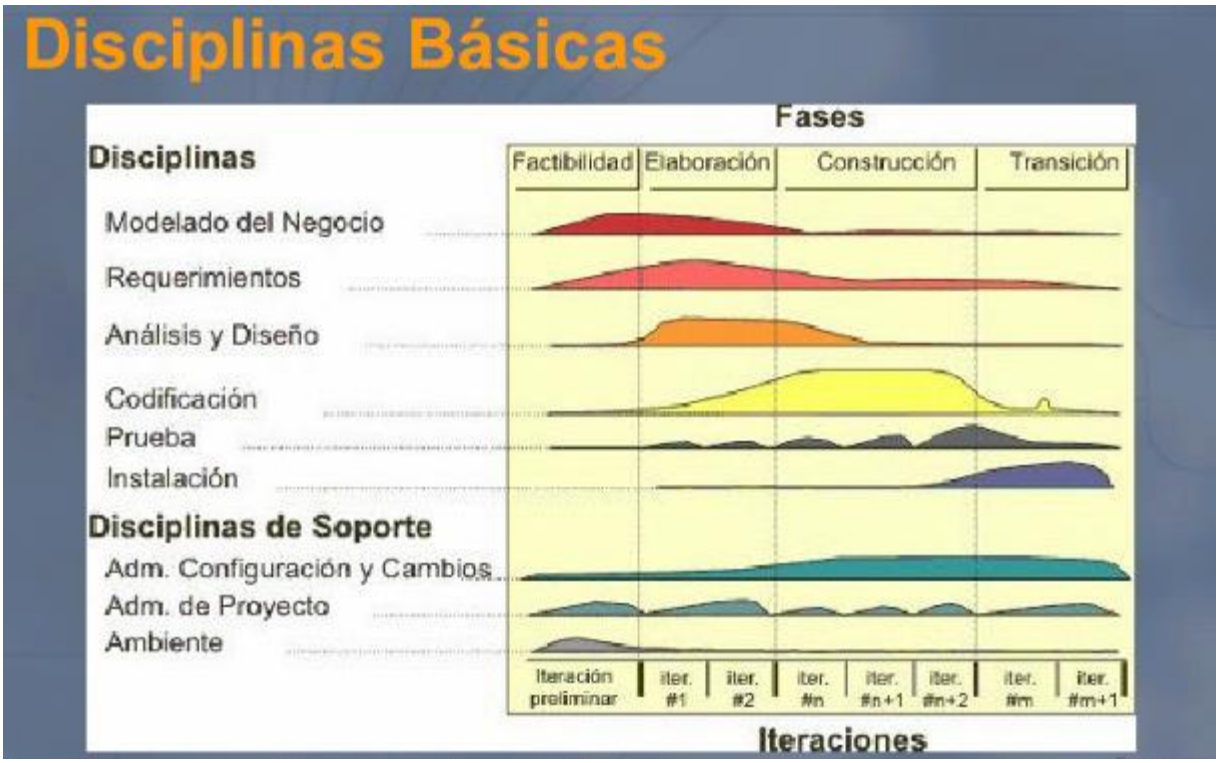
Fases de la Metodología de proyecto RUP

Nota. Fases de la Metodología de proyecto RUP (Torossi, 2002)

Cada fase se subdivide en iteraciones. En cada iteración se desarrolla en secuencia un conjunto de disciplinas o flujos de trabajos.

Figura 5

Disciplinas Básicas (iteraciones) de las Fases de la Metodología RUP

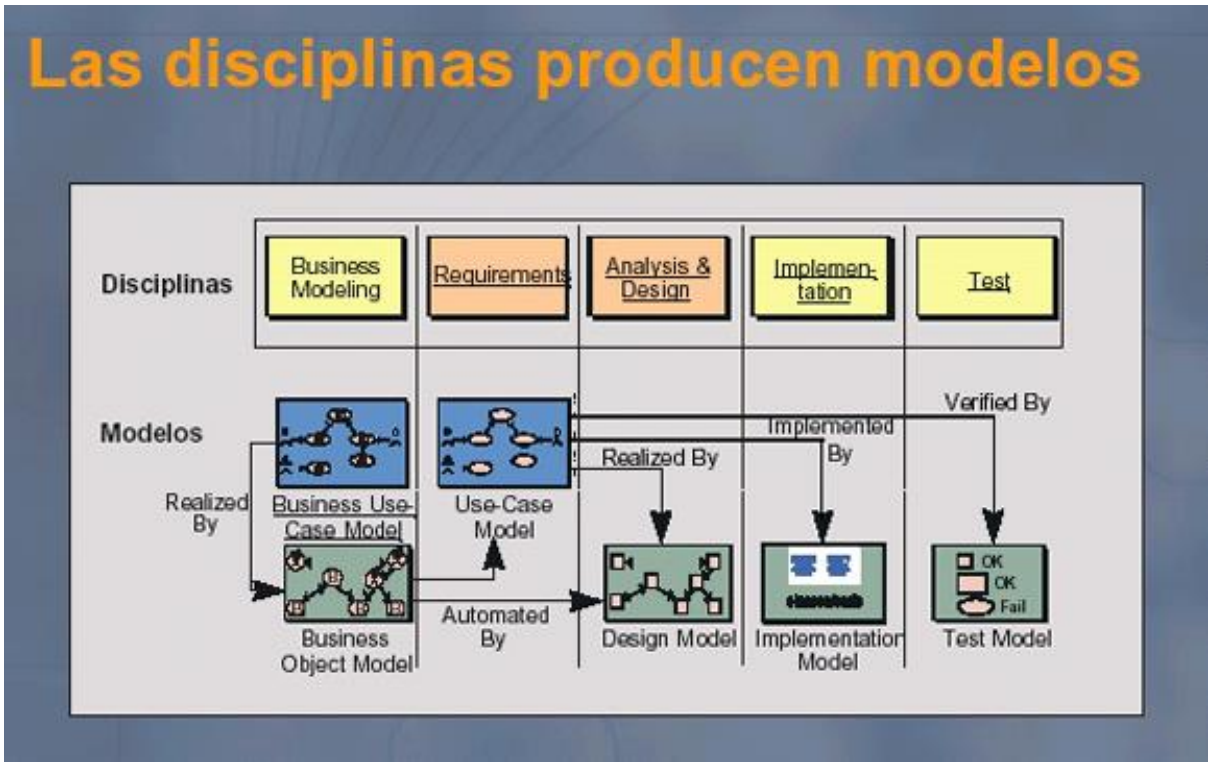


Nota. Disciplinas Básicas (iteraciones) de las Fases de la Metodología RUP (Torossi, 2002)

Cada disciplina es un conjunto de actividades relacionadas (flujos de trabajo) vinculadas a un área específica dentro del proyecto total. Las más importantes son: Requerimientos, Análisis, Diseño, Codificación, y Prueba. Cada disciplina está asociada con un conjunto de modelos que se desarrollan. Estos modelos están compuestos por artefactos. Los artefactos más importantes son los modelos que cada disciplina realiza: modelo de casos de uso, modelo de diseño, modelo de implementación, y modelo de prueba (Torossi, 2002).

Figura 6

Disciplinas que producen modelos - Fases de la Metodología RUP



Nota. Disciplinas que producen modelos - Fases de la Metodología RUP Disciplinas que producen modelos - Fases de la Metodología RUP (Torossi, 2002)

De acuerdo con (López Rosciano & Pech Montejo, 2015) RUP proporciona un entorno controlado para asignar tareas y responsabilidades en su desarrollo. Garantizando un software de alta calidad que satisfaga las necesidades de los usuarios finales aumentando su productividad. La metodología cuenta con las siguientes fases: fase de inicio, fase de elaboración, fase de construcción, fase de transición. En ese sentido según (Torossi, 2002) la metodología RUP se aplica a distintos tipos de sistemas los cuales se basan en componentes que se encuentran conectados mediante interfaces. RUP está orientado a casos de uso, orientado a la arquitectura, es iterativo e incremental. Consta de las siguientes fases: fase de inicio, fase de desarrollo, fase de construcción, fase de transición.

Fases: Cada ciclo constas de cuatro fases: inicio, elaboración, construcción, y transición.

Fase de Inicio

Durante la fase de inicio se debe establecer el modelo de negocio para el sistema y delimitar el alcance del proyecto. Para conseguirlo se deben identificar todas las entidades externas con las que el sistema interactúa (actores) y definir la naturaleza de esta interacción a alto nivel. Esto incluye la identificación de todos los casos de uso y la descripción de algunos importantes. El modelo de negocio incluye el criterio para alcanzar el éxito, el riesgo asumido, la estimación de los recursos necesarios y un plan de fase mostrando los hitos más importantes. Al final de la fase el proyecto puede ser cancelado o replanteado si no se llegan a alcanzar los objetivos (López Rosciano & Pech Montejo, 2015).

Durante la fase de inicio se desarrolla una descripción del producto final, y se presenta el análisis del negocio. Esta fase responde las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema para los usuarios más importantes?
- ¿Cómo podría ser la mejor arquitectura del sistema?
- ¿Cuál es el plan del proyecto y cuánto costará desarrollar el producto?

En esta fase se identifican y priorizan los riesgos más importantes. El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles. Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento en el equipo. Los artefactos que típicamente sobreviven a esta fase son:

- Un enunciado de los mayores requerimientos planteados generalmente como casos de uso.
- Un boceto inicial de la arquitectura.
- Una descripción de los objetivos del proyecto.
- Una versión muy preliminar del plan del proyecto.
- Un modelo del negocio.

La fase de inicio finaliza con el Hito de Objetivos del Ciclo de Vida. Este hito es alcanzado cuando el equipo de proyectos llega a un acuerdo sobre:

- Cuál es el conjunto de necesidades del negocio, y que conjunto de funciones satisfacen estas necesidades.
- Una planificación preliminar de iteraciones.
- Una arquitectura preliminar (Torossi, 2002).

Fase de Elaboración

El propósito de la fase de elaboración es analizar el dominio, establecer la arquitectura, desarrollar el plan de proyecto y eliminar los elementos de riesgo del proyecto. Para cumplir estos objetivos se debe tener una visión global del sistema. Las decisiones arquitectónicas deben ser hechas con conocimiento de todo el sistema: su alcance y la mayoría de requisitos funcionales, no funcionales. En la fase de elaboración, un prototipo ejecutable de la arquitectura es construido en una o más iteraciones, dependiendo del alcance, tamaño, riesgo y grado de innovación del proyecto. El esfuerzo debe al menos abordar los casos de uso críticos identificados en la anterior fase. Al final de la fase de elaboración se encuentra el segundo hito del ciclo de vida de la arquitectura. En este punto se debe examinar detalladamente los objetivos del sistema y el alcance, la elección de la arquitectura y la resolución de la mayoría de riesgos. El proyecto puede ser cancelado o replanteado si no se llegan a alcanzar los objetivos (López Rosciano & Pech Montejo, 2015).

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura. Las iteraciones en la fase de elaboración:

- Establecen una firme comprensión del problema a solucionar.
- Establece la fundación arquitectural para el Software.
- Establece un plan detallado para las siguientes iteraciones.
- Elimina los mayores riesgos.

El resultado de esta fase es la línea base de la arquitectura. En esta fase se construyen típicamente los siguientes artefactos:

- El cuerpo básico del Software en la forma de un prototipo arquitectural.

- Casos de prueba
- La mayoría de los casos de uso (80%) que describen la funcionalidad del sistema.
- Un plan detallado para las siguientes iteraciones.

La fase de elaboración finaliza con el hito de la Arquitectura del Ciclo de Vida. Este hito se alcanza cuando el equipo de desarrollo llega a un acuerdo sobre:

- Los casos de uso que describen la funcionalidad del sistema.
- La línea base de la arquitectura
- Los mayores riesgos han sido mitigados
- El plan del proyecto (Torossi, 2002).

Fase de Construcción

Durante la fase de construcción, todos los componentes y características de la aplicación son desarrolladas e integradas en el producto, y todos los componentes son probados exhaustivamente. La fase de construcción es un proceso que se centra en la gestión de recursos y control de operaciones para optimizar costes, tiempo y calidad. Al final de la fase de construcción se encuentra el tercer hito del ciclo de vida de la arquitectura. Se debe decidir si el Software, los sitios y los usuarios están preparados para ser operativos, sin exponer al proyecto a altos riesgos. Esta versión es llamada beta. La fase de transición puede ser pospuesta por una versión si el proyecto falla al alcanzar sus objetivos (López Rosciano & Pech Montejo, 2015).

Durante la fase de construcción se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo. Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo, puede que no esté libre de defectos. Los artefactos producidos durante esta fase son:

- El Sistema Software
- Los casos de prueba
- Los manuales de usuario

La fase de construcción finaliza con el hito de Capacidad Operativa Inicial. Este hito se alcanza cuando el equipo de desarrollo llega a un acuerdo sobre:

- El producto es estable para ser usado

- El producto provee alguna funcionalidad de valor
- Todas las partes están listas para comenzar la transición (Torossi, 2002).

Fase de Transición

El propósito de la fase de transición es llevar el producto software a la comunidad de usuarios. Una vez el producto se haya entregado al usuario final, surgen problemas que requieren el desarrollo de una nueva versión, la corrección de los problemas o la finalización de las características que fueron pospuestas. Al final de esta fase se encuentra el cuarto hito, el Hito del Producto Entregado. En este punto hay que decidir si los objetivos han sido cumplidos y si se debe empezar otro ciclo de desarrollo (López Rosciano & Pech Montejo, 2015).

La fase de transición cubre el período durante el cual el producto se convierte en la versión beta. Las iteraciones en esta fase continúan agregando características al Software. Sin embargo, las características se agregan a un sistema que el usuario se encuentra utilizando activamente. Los artefactos construidos en esta fase son los mismos que en la fase de construcción. El equipo se encuentra ocupado fundamentalmente en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior. La fase de transición finaliza con el hito de Lanzamiento del Producto. Este hito se alcanza cuando el equipo de desarrollo llega a un acuerdo sobre:

- Se han alcanzado los objetivos fijados en la fase de Inicio.
- El usuario está satisfecho (Torossi, 2002).

2.4. INGENIERÍA DE REQUERIMIENTOS

La ingeniería de requerimientos (IR) cumple un papel primordial en el proceso de producción del software, ya que se enfoca en un área fundamental, la definición de lo que se desea producir, su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin antigüedades, en la forma consistente y compacta las necesidades de los usuarios o clientes de esta manera se pretende minimizar los problemas relacionados por mala gestión de los requerimientos en el desarrollo de sistemas. Los requerimientos de software pueden dividirse en dos categorías requerimientos funcionales y no funcionales (Arias Chaves, 2006).

2.4.1. Requerimientos Funcionales

Los requerimientos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema (Arias Chaves, 2006).

Los Requerimientos de Software son las necesidades de los Stakeholders que requiere que el Sistema deba de cumplir de manera Satisfactoria. Son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema (Valdez Alvarado, 2012).

Por lo tanto (Arias Chaves, 2006) menciona que los requisitos funcionales son las descripciones explicitas del comportamiento que debe tener una solución de software y que información debe manejar. En ese sentido (Valdez Alvarado, 2012) nos indica que los requisitos funcionales son una declaración de cómo debe comportarse un sistema. Define lo que el sistema debe hacer para satisfacer las necesidades o expectativas del usuario. Los requisitos funcionales se pueden considerar como características que el usuario detecta.

2.4.2. Requerimientos no Funcionales

Por otra parte, los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc. (Arias Chaves, 2006).

Características o Cualidades que los Stakeholders esperan como parte del comportamiento del sistema de Software. En ocasiones son orientadas al “Como” en

lugar del “Que”. Las características proveen mucha información acerca de cómo el sistema debe comportase. Están relacionados con las características de calidad del sistema (Valdez Alvarado, 2012).

De acuerdo con (Arias Chaves, 2006) se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace. Por otro lado, según (Valdez Alvarado, 2012) los requisitos no funcionales se originan en la necesidad del usuario, debido a restricciones presupuestarias, políticas organizacionales, la necesidad de interoperabilidad con otros sistemas de software o hardware, o factores externos tales como regulaciones de seguridad, políticas de privacidad, entre otros.

2.5. PRUEBAS DE SOFTWARE

2.5.1. Caja Blanca

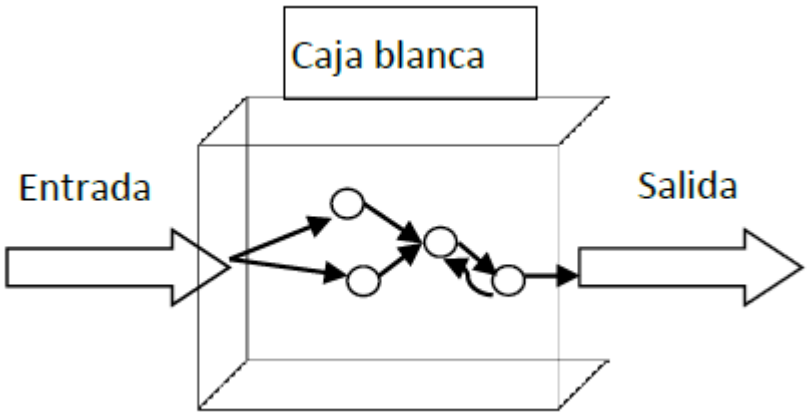
La técnica de caja blanca, a veces definida como prueba de “caja de cristal” o “caja transparente”, es una técnica de diseño de casos de prueba que usa la estructura de control para obtener los casos de prueba. Dentro de esta estructura de control podemos encontrar la estructura de un componente de software como puede ser sentencias de decisiones, caminos distintos del código, la estructura de una página web, etc. Los métodos de prueba de caja blanca aportan los siguientes puntos: garantizar que todas las rutas del código se revisan al menos una vez, revisan las condiciones lógicas, revisan estructuras de datos (Sanchez Peño, 2015).

La prueba de caja blanca, denominadas a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control de diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero de software puede obtener casos de prueba que: garanticen que se ejercite por lo menos una vez todos los caminos independientes de cada módulo, ejecuten todas las decisiones lógicas en sus vertientes verdadera y falsa, ejecuten todos los bucles en sus límites y con sus vertientes verdadera y falsa,

ejerciten las estructuras internas de datos para asegurar su validez (Callisaya Corina, 2015).

Figura 7

Enfoque de diseño de pruebas de Caja Blanca



Nota. Enfoque de diseño de pruebas de Caja Blanca

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

Cálculo de la Complejidad Ciclomática

Recordemos que la Complejidad Ciclomática de un grafo de flujo V (G) establece el número de caminos independientes o caminos básicos. Este puede calcularse de tres formas distintas, una de ellas es la siguiente:

$$V(G)=A-N+2,$$

Donde A es el número de aristas y N es el número de nodos.

En ese sentido según (Sánchez Peño, 2015) Caja blanca es a veces denominada prueba de "caja de cristal" o "caja transparente", es una técnica que utiliza estructuras de control las cuales brindan el poder garantizar que todas las rutas de código se revisen al menos una vez. De acuerdo con (Callisaya Corina, 2015) las pruebas de caja blanca, son un método que utiliza una estructura de control de diseño para derivar casos de prueba con lo cual se puede lograr asegurarse que se ejecute todas las decisiones lógicas en sus lados verdadero y falso.

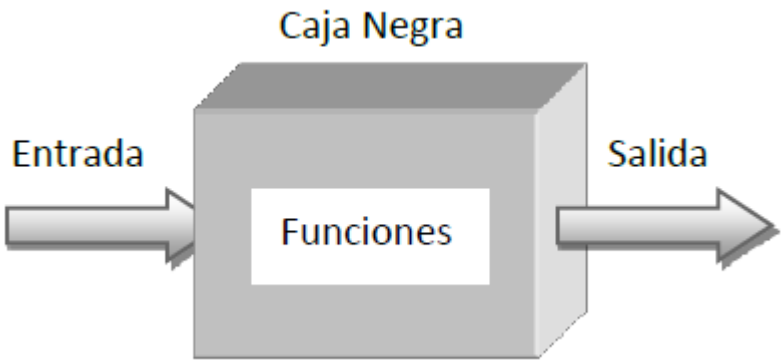
2.5.2. Caja Negra

Las técnicas de diseño de caja negra, también llamadas pruebas de comportamiento, son las que utilizan el análisis de la especificación, tanto funcional como no funcional, sin tener en cuenta la estructura interna del programa para diseñar los casos de prueba y, a diferencia de caja blanca, estas pruebas se suelen realizar durante las últimas etapas de la prueba. Con los métodos de caja negra se intenta encontrar los errores: funciones incorrectas o faltantes, errores de inicialización y terminación, errores de interfaz, errores en las estructuras (Sanchez Peño, 2015).

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que: las funciones del software son operativas, la entrada se acepta de forma correcta, se produce una salida correcta, la integridad de la información externa se mantiene. Las pruebas de caja negra pretenden encontrar estos tipos de errores: funciones incorrectas o ausentes, errores en la interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación (Callisaya Corina, 2015).

Figura 8

Enfoque de diseño de pruebas de Caja Negra



Nota. Enfoque de diseño de pruebas de Caja Negra

Por lo tanto (Sánchez Peño, 2015) menciona que Caja negra es también conocida como prueba de comportamiento, utilizan un análisis prescriptivo, estas pruebas generalmente se realizan más tarde para encontrar errores, funciones incorrectas, errores de inicialización y terminación. En ese sentido (Callisaya Corina, 2015) nos indica que la prueba de caja negra puede definirse como una técnica donde se busca la verificación de las funcionalidades del software o aplicación analizada y demostrar que la funcionalidad del software funciona, que las entradas se aceptan correctamente, que las salidas se producen correctamente y que se mantiene la integridad de la información externa.

2.6. SEGURIDAD

Los sistemas de seguridad críticos son sistemas en los que es esencial que el funcionamiento del sistema sea siempre seguro. Esto es, el sistema nunca debería provocar daños en las personas o en el entorno del sistema incluso si éste falla (Sommerville, 2005).

La seguridad del software es una actividad del aseguramiento del software que se centra en la identificación y evaluación de los peligros potenciales que podrían afectarlo negativamente y que podrían ocasionar que falle todo el sistema. Si los peligros se identifican al principio del proceso del software, las características de su diseño se especifican de modo que los eliminen o controlen

a) Seguridad a nivel de base de datos:
    - El acceso a la base de datos
    -Trazabilidad de la Información
b) Seguridad a nivel de aplicación:
    - Control de acceso
    - Validación de datos de entrada
    - Pruebas de código
c) Copias de seguridad:
    - Backup de la base de datos (Pressman, 2010).

En ese sentido según (Sommerville, 2005) la seguridad en un sistema es complicado, un sistema no debe generar problemas en las personas o en su entorno. De acuerdo con (Pressman, 2010) la seguridad de un sistema está destinada a identificar y evaluar amenazas. Si las amenazas se reconocen temprano en el proceso del software para prevenirlos o controlarlos.

2.6.1. Seguridad a nivel de Base de Datos

Es la capacidad del sistema para proteger Datos, Servicios y Recursos de usuarios no autorizados. El fin de la seguridad es garantizar la protección o estar libre de todo peligro y/o daño, y que en cierta manera es infalible.

- **Confidencialidad:** Nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades.
- **Integridad:** Significa que los objetos solo pueden ser modificados por elementos autorizados, y de una manera controlada.
- **Disponibilidad:** Indica que los objetos del sistema tienen que permanecer accesibles a elementos autorizados.

2.6.2. *Seguridad a nivel de aplicación*

- Control de acceso
- Validación de datos de entrada
- Pruebas de código

2.6.3. Copia de Seguridad

Se conoce con este nombre al resultado de efectuar una copia de todos, o algunos archivos, que se encuentran en el medio de almacenamiento de una o varias computadoras en otros medios diferentes a este último, para poder recuperarlos en otro momento si se pierden o se dañan Los archivos originales (Alejandro Montoya, 2000).

Las copias de seguridad son mecanismos de recuperación que poseen los administradores para rescatar información perdida; es fundamental en la planificación de la seguridad se dan cuando no se realizan las mismas. Esto sucede de manera frecuente y el resultado puede ser desastroso (Rodas Palomeque & Ulloa Brito, 2006).

En ese sentido según (Alejandro Montoya, 2000) la copia de seguridad es el resultado al crear todos o ciertos archivos. De acuerdo con (Rodas Palomeque y Ulloa Brito, 2006) la copia de seguridad es un mecanismo de recuperación que los administradores necesitan para guardar información; esto sucede en ausencia de un plan de seguridad.

2.6.4. Autenticación

Consiste en un sistema para certificar que el usuario es quien dice ser; lo más común es utilizar una combinación de identificador de usuario único y contraseña, aunque existen otros. Un sistema de single-sign-on consiste, por tanto, en un protocolo de autenticación que funcione en más de un sistema; es federado si el sistema responsable de la autenticación puede ser cualquiera que cumpla el estándar definido por el sistema; por otro lado, será un sistema delegado si el sistema que autentica es uno predeterminado (Sánchez, 2011).

Toda organización que se basa en un sistema de información tiene que garantizar el proceso de conexión a los sistemas y aplicaciones. La creación de una fuente única y fiable de las identidades, asociada a la gestión de los derechos son los dos pilares de una buena infraestructura de gestión de las identidades y accesos (Evidian, 2011).

En ese sentido según (Sánchez, 2011) la autentificación de un usuario usa una combinación única de ID de usuario y contraseña, autenticación puede ser cualquier sistema que cumpla con los estándares establecidos por el sistema. De acuerdo con (Evidian, 2011) cualquier entorno basado en un sistema debe permitir establecer una única fuente de identidad confiable a eso se llamaría autentificación.

2.6.5. Roles y permisos

Estos son privilegios que permiten a los usuarios accionar sobre los objetos de la base de datos. Los objetos de la base de datos son elementos que pueden ser aplicados a la protección de la seguridad. Al asignar un rol a un inicio de sesión, y a una cuenta de seguridad de usuario, se otorgan permisos para desarrollar determinadas tareas, tanto sobre el servidor, como en la base de datos. Además de los permisos otorgados directamente a los usuarios, se puede asignar permisos mediante roles. Existen dos colecciones de roles fijos (roles establecidos por el gestor), los del servidor y los de la base de datos. También los gestores permiten la creación de roles por parte de los usuarios: con estos se pueden asignar permisos precisos, necesarios para una aplicación (Rodas Palomeque & Ulloa Brito, 2006).

El OJS trabaja a través de roles de usuario los cuales están encargados de las diversas tareas en el proceso editorial, desde la creación de la revista, el ingreso de artículos hasta su publicación. Cada rol cuenta con diferentes niveles de acceso e interacción con el sistema y/o con la publicación (Chavez Gonzales, 2012).

Por lo tanto (Rodas Palomeque & Ulloa Brito, 2006) menciona que un rol es un conjunto de permisos, los cuales definen una actividad concreta que puede realizar un usuario. Un usuario puede tener asignado más de un rol. En ese sentido (Chávez Gonzales, 2012) nos indica que un rol es una colección de permisos definida para que todo el sistema pueda asignar a usuarios específicos en contextos específicos.

2.6.6. Cifrado de datos

El cifrado es un mecanismo que protege tu información importante, como documentos, fotografías o transacciones en línea, del acceso o modificación por personas no autorizadas. El cifrado funciona utilizando un “algoritmo” (fórmula

matemática) y una llave, para convertir datos legibles (texto plano) a una forma que otros no puedan entender (texto cifrado). El algoritmo es la receta general para el cifrado y tu llave hace únicos los datos cifrados – sólo personas con tu llave única y el mismo algoritmo pueden descifrarlo. Las llaves generalmente son una larga secuencia de números protegidos por un mecanismo común de autenticación como contraseñas, tokens o biométricos, como tu huella digital (Sans Securing The Human, 2011).

Cifrar o encriptar datos significa alterarlos, generalmente mediante el uso de una clave, de modo que no sean legibles para quienes no posean dicha clave. Luego, a través del proceso de descifrado, aquellos que sí poseen la clave podrán utilizarla para obtener la información original. Esta técnica protege la información sensible de una organización, ya que, si los datos cifrados son interceptados, no podrán ser leídos (ESET Enjoy Safer Technology, 2014).

De acuerdo con (Sans Seusing The Human, 2011) el cifrado es la conversión de datos de un formato legible a un formato codificado. Los datos cifrados solo se pueden leer o procesar información. Por otro lado, según (ESET Enjoy Safer Technology, 2014) cifrar significa reemplazarlos, es un método de protección de datos que consiste en alterarlos hasta hacerlos ilegibles. Los datos pasan de ser texto sin formato a ser texto cifrado por medio de un método denominado algoritmo.

2.7. MÉTRICAS DE CALIDAD

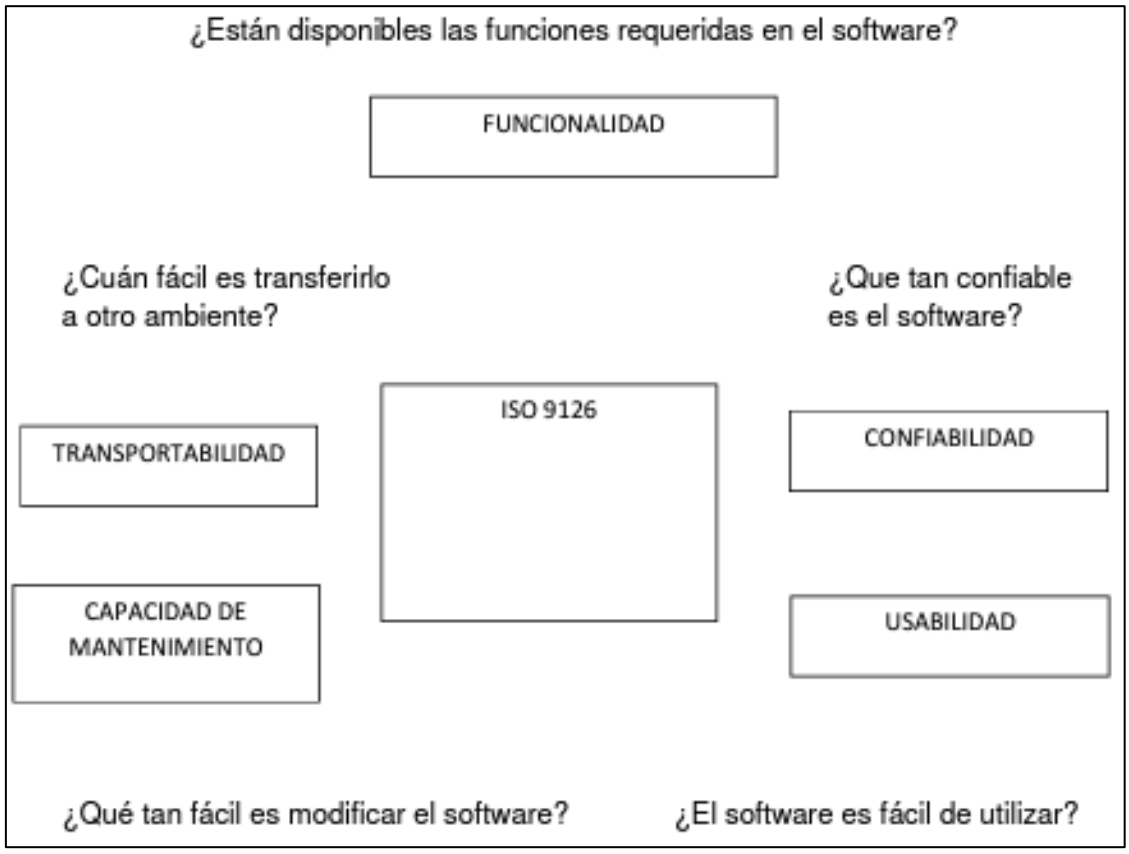
2.7.1. Norma ISO/IEC 9126

La organización internacional para la estandarización ISO fue creada en 1946 con el fin de facilitar el comercio internacional, la coordinación internacional y la unificación de estándares industriales promoviendo una serie de simples patrones de estándares que deberán ser reconocidos y respetados (grupo de investigación Praxiom). ISO 9126 fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de la calidad del software y así refinarlo en un periodo de 10 años. Muchos estudios criticaron la ISO 9126 por no recomendar requerimientos específicos de calidad en vez de definir un esquema general para la evaluación de calidad del software. Nosotros creemos que este es de hecho una de las fortalezas y así es más adaptable y puede ser usado a través de varios sistemas incluso sistemas

de aprendizaje virtual. El producto original definió seis características del producto estas seis características son divididas en un numero de sub -características.

Figura 9

División de la Norma ISO/IEC 9126



Nota. División de la norma ISO/IEC 9126 (Facultad de Tecnología de Información, 2002)

Tabla 10

Características y subcaracterísticas ISO-9126

Características Propuestas por ISO-9126		
Funcionalidad	Conveniencia Precisión Interoperabilidad Seguridad	✓ Puede el software desempeñar las tareas requeridas. ✓ ¿El resultado es el esperado?

		✓ ¿el sistema puede interactuar con otro? ✓ ¿el sistema impide el acceso no autorizado?
Confiabilidad	Vencimiento Tolerancia a las fallas Capacidad de recuperación.	✓ ¿muchas de las fallas han sido eliminadas durante el tiempo? ✓ ¿el software es capaz de manejar errores? ✓ ¿Puede el software reasumir el funcionamiento y restaurar datos perdidos después de la falla?
Utilidad	Claridad Capacidad de aprendizaje Operatividad atractivo	✓ ¿el usuario comprende fácilmente como usar el sistema? ✓ ¿Puede el usuario aprender fácilmente a utilizar el sistema? ✓ ¿el usuario puede utilizar el sistema sin mucho esfuerzo? ✓ ¿la interfaz se ve bien?
Capacidad de Mantenimiento	Capacidad de análisis Variabilidad Estabilidad Capacidad de prueba	✓ ¿las fallas Pueden ser fácilmente diagnosticadas? ✓ ¿el sistema puede ser fácilmente modificado? ✓ ¿el sistema puede seguir funcionando si se hacen cambios? ✓ ¿el sistema puede ser probado fácilmente?

Transportabilidad	Adaptabilidad Capacidad de instalación Conformidad Capacidad para remplazar	✓ ¿El software se puede trasladar a otros ambientes? ✓ ¿el software se puede instalar fácilmente? ✓ ¿el software cumple con los estándares de Transportabilidad? ✓ ¿el software puede remplazar fácilmente otro software?
Todas las características	Cumplimiento	✓ ¿el software cumple con todas las leyes y reglamentos?

Nota. Características y subcaracterísticas ISO-9126 (Facultad de Tecnología de Información, 2002)

Estas características y sub -características representan un modelo detallado para la evaluación de cualquier sistema de software (Facultad de tecnologia de la Información, 2002).

Funcionalidad: En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.

Confiabilidad: Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

Usabilidad: Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

Mantenibilidad: Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.

Portabilidad: En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro (Abud Figueroa, 2012),

Por lo tanto (Facultad de tecnología de la Información, 2002) menciona que la norma ISO 9126 es un estándar internacional para la evaluación de la calidad del software. En ese sentido (Abud Figueroa, 2012) nos indica que la norma ISO 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimiento, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

2.8. COSTOS

2.8.1. COCOMO II

COCOMO intermedio calcula el esfuerzo y el coste en función del Tamaño estimado del programa y de un conjunto de "guías de coste" que incluyen una evaluación subjetiva del producto, hardware, personal y atributos del producto.

Formula de esfuerzo:

$$E = a * KLDC^{b} * \prod_{i=1}^{n=15} M(x_i)$$

Dónde:

y b, representan los coeficientes COCOMO nivel intermedio

E, es el esfuerzo aplicado en personas−mes,

KLDC, es el número estimado de Líneas de Código expresadas en miles distribuidas para el proyecto.

$\prod_{i=1}^{n=15} M(x_i)$, representa la productoria de los 15 factores de costos (Cost-Drives) evaluados para el proyecto.

Donde E es el esfuerzo aplicado en personas−mes, KLDC es el número estimado de Líneas de Código (en miles) distribuidas para el proyecto.

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.

- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura. Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Esta diversidad podría representarse con el siguiente esquema (María del C. Lopez & Alejandra Otazú, 2013).

Una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. COCOMO II, este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto. COCOMO define tres modos de desarrollo o tipos de proyectos:

Figura 10

Valores constantes COCOMO Intermedio

PROYECTO	a	b	c	d
ORGÁNICO	2.40	1.05	2.50	0.38
SEMIACOPLADO	3.00	1.12	2.50	0.35
EMPOTRADO	3.60	1.20	2.50	0.32

Nota. Valores constantes COCOMO Intermedio

- Orgánico: proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- Semiacoplado: proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- Empotrado: proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad (Fernandez Mamani, 2016).

De acuerdo con (María del C. López & Alejandra Otazú, 2013) COCOMO II se compone de tres modelos: Composición de aplicaciones, Diseño temprano y Arquitectura tardía. Han surgido en respuesta a la diversidad del mercado de desarrollo de software actual y futuro. Por otro lado, según (Fernández Mamani, 2016) una de las tareas más importantes en la planificación de proyectos de software es la estimación, que incluye hasta cierto punto determinar los recursos, costos, tiempo y esfuerzo de hardware y software necesarios para su desarrollo. En COCOMO II, el modelo permite realizar estimaciones basadas en el tamaño del software, y se incorpora un conjunto de factores de costo y escala.

2.9. HERRAMIENTAS

2.9.1. MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocidos y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. MySQL está disponible para múltiples plataformas, la seleccionada para los ejemplos de este libro es GNU/Linux. Sin embargo, las

diferencias con cualquier otra plataforma son prácticamente nulas, que la herramienta utilizada en este caso es el cliente mysql-client, que permite interactuar con un servidor MySQL (local o remoto) en modo texto. De este modo es posible realizar todos los ejercicios sobre un servidor instalado localmente o, a través de Internet, sobre un servidor remoto (Casillas Santillan et al., 2014).

MySQL es el sistema de administración de bases de datos (Database Management System, DBMS) más popular, desarrollado y proporcionado por MySQL AB. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interactuación con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.
También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.
Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

Características MySQL:

- Velocidad. MySQL es rápido.
- Facilidad de uso. Es un sistema de base de datos de alto rendimiento, pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes.
- Coste. Es gratuito.
- Capacidad de gestión de lenguajes de consulta. MySQL comprende SQL, el lenguaje elegido para todos los sistemas de bases de datos modernos.
- Capacidad. Pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias bases de datos simultáneamente. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java, PHP y Python.

- Conectividad y seguridad. MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar de Internet. Dispone de control de acceso.
- Portabilidad. MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes, así como bajo Microsoft Windows.
- Distribución abierta. Puede obtener y modificar el código fuente de MySQL (Pérez Garcia, 2007).

De acuerdo con (Casillas Santillan et al., 2014) MySQL es un sistema de gestión de bases de datos relacional, conocido por su simplicidad y buen rendimiento. Tiene un alto grado de solidez y rápido desarrollo. MySQL está disponible para varias plataformas. Por otro lado, según (Pérez García, 2007) MySQL es el sistema multiproceso y multiusuario, se adapta a diferentes entornos de desarrollo, es de uso gratuito e incluso de modificación gratuita. Es rápida, posee alto rendimiento, es relativamente simple y mucho más fácil de configurar.

2.9.2. PHP

PHP es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML. En este libro se explicará en detalle la sintaxis y el funcionamiento de este lenguaje, de momento se realiza a continuación una breve comparativa con las otras tecnologías del lado del servidor descritas previamente.

Comparando el lenguaje PHP con el lenguaje Perl, utilizado habitualmente en la programación CGI, puede decirse que PHP fue diseñado para desarrollo de scripts orientados a web, mientras que Perl fue diseñado para hacer muchas más cosas y debido a esto, se hace muy complicado. La sintaxis de PHP es menos confusa y más estricta, pero sin perder la flexibilidad. En definitiva, PHP es uno de los lenguajes más utilizados actualmente en el desarrollo de aplicaciones web y viene experimentado un constante crecimiento en su nivel de utilización en Internet. Este libro trata de

humildemente contribuir a continuar con el proceso de difusión de esta tecnología (Cobo et al., 2005).

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, bucles y funciones, cercano a C o a JavaScript. No es un lenguaje de marcas como podría ser HTML, XML o WML. PHP (acrónimo de "Hypertext Preprocessor") es un lenguaje "open source" interpretado de alto nivel embebido (introducido) en páginas HTML y ejecutado en el servidor. Es decir, lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Por ejemplo, al acceder a una página escrita en PHP, el cliente solamente recibirá el resultado de la ejecución de esta en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es decir es independiente del navegador, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluido HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse también como procesador CGI.
Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP. Las siguientes bases de datos son algunas de las cuales están soportadas actualmente: PostgreSQL, dBase, MySQL, Oracle, ODBC (Pelissier Q., 2002).

En ese sentido según (Cobo et al., 2005) PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web, se caracteriza por sus funciones, versatilidad, durabilidad y modularidad. De acuerdo con (Pelissier Q., 2002)

PHP es un lenguaje de programación de estilo clásico, es un lenguaje de "código abierto", PHP es un lenguaje del lado del servidor, la característica más potente y significativa de PHP puede ser que admite una gran cantidad de bases de datos.

2.9.3. HTML

HTML son las iniciales de Hiper Text Markup Language. Es un conjunto o serie de etiquetas incluidas en archivos de texto que definen la estructura de un documento WWW y sus vínculos con otros documentos. Los navegadores WWW leen estos archivos de texto e interpretan esas etiquetas para determinar cómo desplegar la página Web (Anibarro Zelaya, 2001).

HTML es un lenguaje que se utiliza para la creación de páginas en la WWW. Por página entenderemos el documento que aparece en el visualizador o navegador. HTML se compone de una serie de comandos, que son interpretados por el visualizador, o programa que utilizamos para navegar por el WWW. En última instancia es el visualizador el que ejecuta todas las ordenes contenidas en el código HTML, de forma que un visualizador puede estar capacitado para unas prestaciones, pero no para otras. Así, podremos especificar que una página tenga una imagen de fondo, o un texto parpadeando, pero si nuestro visualizador no está capacitado para esas funciones, no podremos verlas (Area de Tecnologias de la Informacion y las Comunicaciones Aplicadas, 2010).

Por lo tanto (Anibarro Zelaya, 2001) menciona que HTML significa lenguaje de marcado de hipertexto, se encuentra contenida en un archivo texto por una serie de etiquetas. En ese sentido el (Área de Tecnologías de la Información y las Comunicaciones Aplicadas, 2010) nos indica que HTML es el lenguaje utilizado para crear páginas en la WWW, el documento que se muestra en el visor o navegador.

2.9.4. JavaScript

JavaScript se introdujo en 1995 como una forma de agregar programas a páginas web en el navegador Netscape Navigator. El lenguaje ha sido desde entonces adoptado por todos los otros navegadores webs principales. Ha hecho que las aplicaciones web modernas sean posibles: aplicaciones con las que puedes

interactuar directamente, sin hacer una recarga de página para cada acción. JavaScript también es utilizado en sitios web más tradicionales para proporcionar diversas formas de interactividad e ingenio. Es importante tener en cuenta que JavaScript casi no tiene nada que ver con el lenguaje de programación llamado Java. El nombre similar fue inspirado por consideraciones de marketing, en lugar de buen juicio. Cuando JavaScript estaba siendo introducido, el lenguaje Java estaba siendo fuertemente comercializado y estaba ganando popularidad (Haverbeke, 2018).

JavaScript es un lenguaje interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web (Gauchat, 2012).

De acuerdo con (Haverbeke, 2018) JavaScript es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs, JavaScript no tiene relación con el lenguaje de programación Java. Por otro lado, según (Gauchat, 2012) JavaScript se considera un complemento. JavaScript se ha ampliado para la portabilidad y la integración.

2.9.5. CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (Eguiluz Pérez, 2008).

CSS es la abreviatura de Cascade Style Sheets (Hojas de Estilo en Cascada) y se trata de un lenguaje de texto que se incrusta en las páginas web para modificar el formato de la página. Actúa sobre HTML haciendo que las etiquetas HTML se muestren en el navegador con el formato que se indique. Es capaz de actuar sobre todas las etiquetas del mismo tipo o sobre unas concretas. Se puede almacenar en un archivo aparte que después se puede usar para varias páginas a la vez. De modo que, si cambiamos algo en el estilo, al instante se reflejará en todas las páginas.
CSS por lo tanto facilita la homogeneidad de las páginas y su mantenimiento. Hoy en día se considera una técnica imprescindible para dar formato a las páginas web. Además, se puede aplicar también a código XML (Sanchez Asenjo, 2013).

En ese sentido según (Eguiluz Pérez. 2008) CSS es un lenguaje de hojas de estilo en cascada usado para definir la presentación de un documento estructurado escrito en HTML, el cual mejora la accesibilidad de los documentos, reduce la complejidad del mantenimiento y permite visualizar un mismo documento en un

número ilimitado de dispositivos diferentes. De acuerdo con (Sanchez Asenjo, 2013) CSS un lenguaje de texto, contribuye a la uniformidad de la página y su mantenimiento.

2.9.6. Laravel

Laravel es un framework web de código abierto utilizado para desarrollar aplicaciones web y sistemas web. Laravel está programado en PHP. Como se comentó anteriormente, Laravel busca que el desarrollo de las aplicaciones se haga de una forma elegante y simple. Gran parte de Laravel está formado por dependencias de otros sistemas, como Sinfony, por los que su desarrollo es dependiente de estas dependencias (Sanchez Pedros, 2017).

Laravel es el nombre de un framework creado para trabajar con PHP creado en el año 2011 por Taylor Otwell y que, con el paso del tiempo, ha ido ganando terreno a otros framework para trabajar con PHP como Symfony o Zend Framework. Se trata de framework de desarrollo con una curva de aprendizaje muy rápida y que maneja una sintaxis expresiva, elegante, con el objetivo de eliminar la molestia del desarrollo web facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y caché. Proporciona potentes herramientas necesarias para construir aplicaciones robustas y que puede ser utilizado tanto para proyectos a nivel empresarial como para proyectos más sencillos, lo que significa que es perfecto para todos los tipos de proyectos (Hostalia Whitepapers, 2003).

Por lo tanto (Sánchez Pedros, 2017) menciona que Laravel es una plantilla, un esquema conceptual que simplifica la elaboración de un código más amplio el cual estas escrito en PHP, hace que el desarrollo de aplicaciones sea elegante y simple. En ese sentido (Hostalia Whitepapers, 2003) nos indica que Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación con una sintaxis expresiva y elegante y tiene como objetivo simplificar tareas comunes como la autenticación, el enrutamiento.

2.9.7. Boostrap

Bootstrap es un framework CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen

tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Aunque el desarrollo del framework Bootstrap fue iniciado por Twitter, fue liberado bajo licencia MIT en el año 2011 y su desarrollo continúa en un repositorio de GitHub.

Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías. Desde la aparición de Bootstrap, el framework se ha vuelto bastante más compatible con desarrollo web responsivo (González González & Galarza Galarza, 2016).

Bootstrap es un Framework para desarrollo de interfaces de usuario que fue desarrollado por 2 de los desarrolladores de Twitter para asegurarse de que los proyectos tuvieran una apariencia consistente. Sin embargo, muchas personas decían ¿no es otro Framework para Interfaces de Usuario? Ya tenemos jQuery UI, Kendo UI y un millón de otros más, ¿Por qué necesitamos otro? Bootstrap es diferente. Su propósito no es crear ventanas de dialogo o efectos de desplazamiento (aunque realmente lo hace bastante bien), su propósito tampoco es que se puedan aplicar temas a nuestros sitios Web. Bootstrap está diseñado para ayudar a las personas que no son diseñadoras, a equilibrar sus plantillas y diseños.

La parte interesante, es que ha sido desarrollado por desarrolladores y para desarrolladores. En la mayoría de los casos todo lo que se necesita es añadir una clase o algunos atributos de datos HTML.

En este módulo describiremos qué es el Framework Bootstrap, explicaremos la estructura de archivos y cada uno de los componentes que forman parte de él, además, explicaremos el sistema Grid predeterminado de Bootstrap el cual nos ayudará a crear un sitio web responsivo con mayor facilidad (Huerta de los Santos & Muñoz Serafin, 2018).

De acuerdo con (González González & Galarza Galarza, 2016) Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

Bootstrap es una herramienta para crear interfaces de usuario limpias, proporciona las herramientas necesarias para crear sitios web utilizando estilos y elementos de su biblioteca. Por otro lado, según (Huerta de los Santos y Muñoz Serafín, 2018) Bootstrap es una herramienta que permite crear una interfaz web, a través de lenguajes de programación como HTML, CSS o JavaScript; los cuales permiten adaptar el contenido de la web a los diferentes tipos de pantallas (diseño responsive) y mejorando, por tanto, la experiencia de usuario.

CAPÍTULO III

CAPÍTULO III
MARCO APLICATIVO

3.1. INTRODUCCIÓN

En este capítulo se desarrollará la fase de diagnóstico de la situación actual de la cooperativa mediante el uso de técnicas de ingeniería de requerimientos, análisis y diseño del sistema propuesto, el cual permitirá lograr los objetivos de una forma adecuada y en un determinado periodo de tiempo, también se identificarán los principales actores que de una u otra manera se verán beneficiados.

Luego de detallar los elementos centrales del problema expuesto en el Marco Preliminar, también se considerará lo expuesto en el Marco Teórico los cuales se deben poner en marcha para aplicarlos en el proyecto. Haciendo énfasis a todo lo que significa el proceso de desarrollo de software; desde los requerimientos del usuario hasta la prueba final del sistema. Cada una de las actividades que son necesarias para la transformación de los requisitos del usuario en un sistema software, serán detalladas de acuerdo a la metodología utilizada RUP.

3.2. FASE DE INICIO

En esta fase se define y delimita el alcance del proyecto, identificando los riesgos asociados al proyecto y proponer una visión general de la estructura del software a desarrollarse. Durante el proceso de esta fase, se mostrará el modelado de negocio y los requerimientos del sistema.

3.2.1. Análisis de la situación actual

La Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. se dedica a fomentar el ahorro y otorgar a sus socios recursos financieros en calidad de préstamo. Actualmente no cuenta con un Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad.

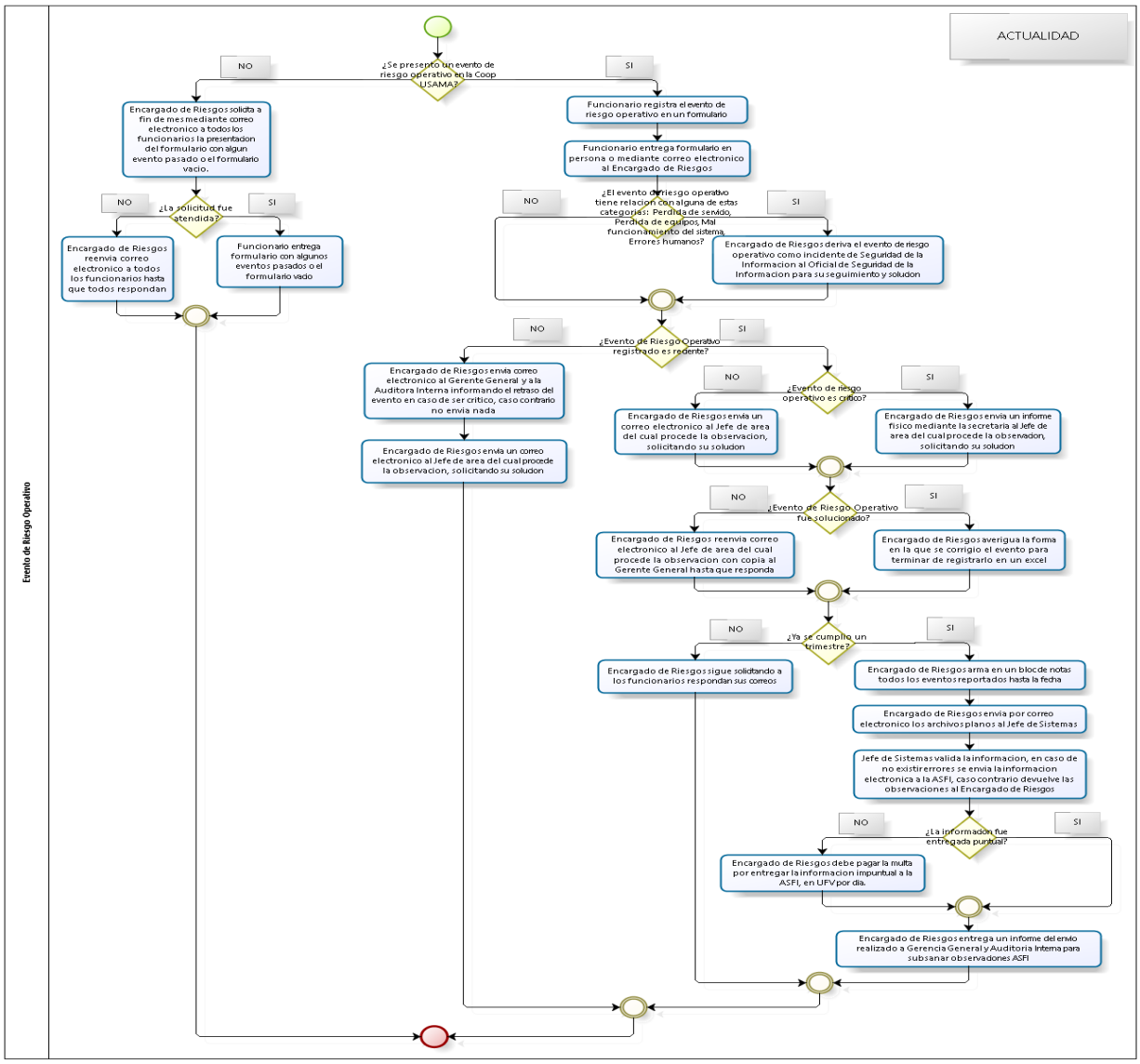
Los registros se realizan de forma manual, cuando ocurre un Evento de Riesgo Operativo o Incidente de Seguridad de la Información, el funcionario lo registra en un formulario y lo entrega al Encargado de Riesgos u Oficial de Seguridad de la Información, acción que conlleva mucho tiempo, desorganización, acumulación y descuido, la falta de un software para el registro de eventos o incidentes es observado

por el área de Auditoría Interna de la Cooperativa y por lo tanto es objeto de observación por el ente regulador ASFI.

La Oficina Central de la Cooperativa se encuentra ubicada en la avenida Juan Pablo II No. 2805 de la Zona de Ferropetrol, altura Cruz Papal lado Cristembo de la ciudad de El Alto.

Figura 11

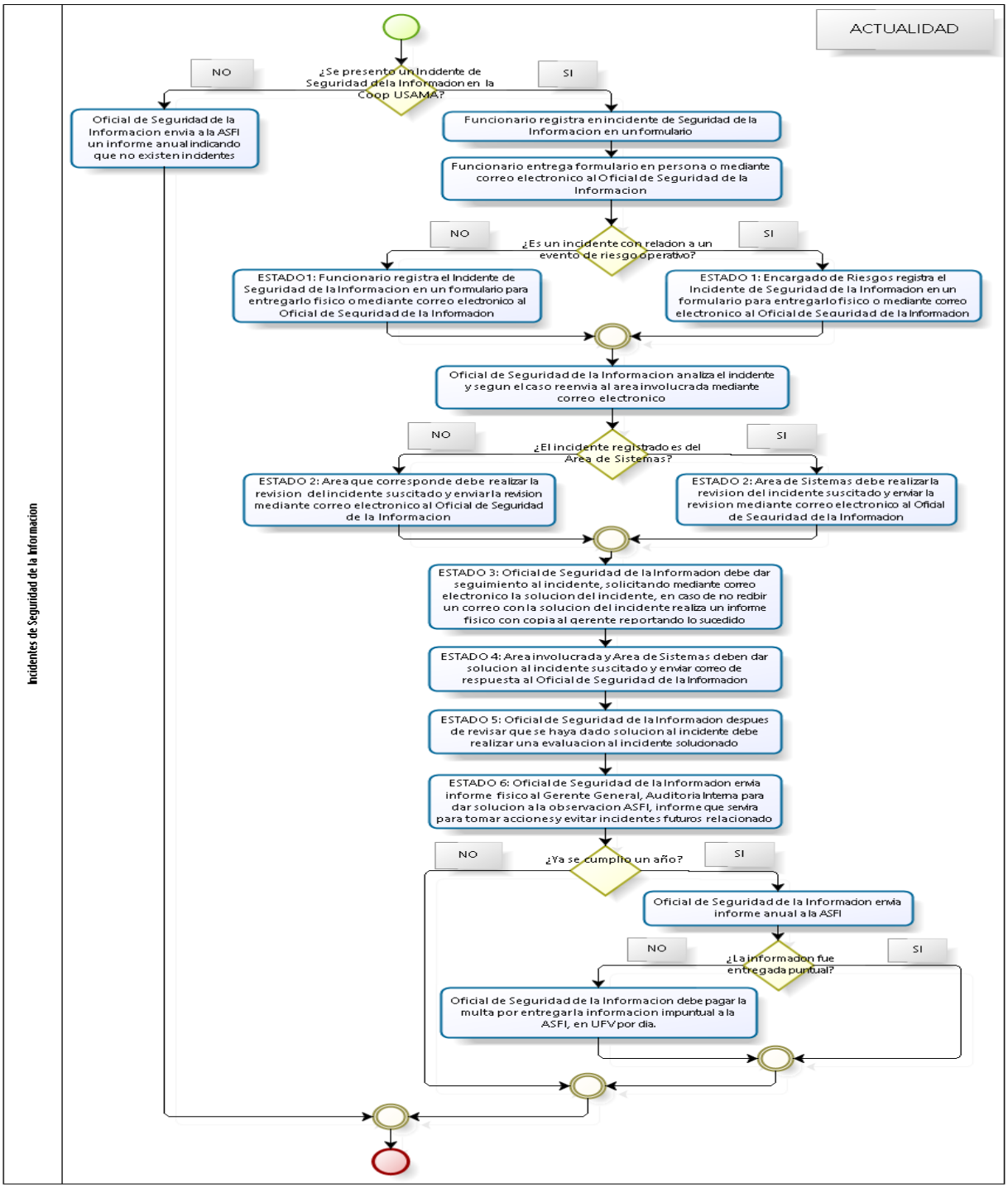
Diagrama de Flujo - Evento de Riesgo Operativo (Actualmente)



Nota. Diagrama de Flujo – Evento de Riesgo Operativo, cómo se maneja dentro de la Cooperativa Unión Santiago de Machaca USAMA Ltda. (Actualmente).

Figura 12

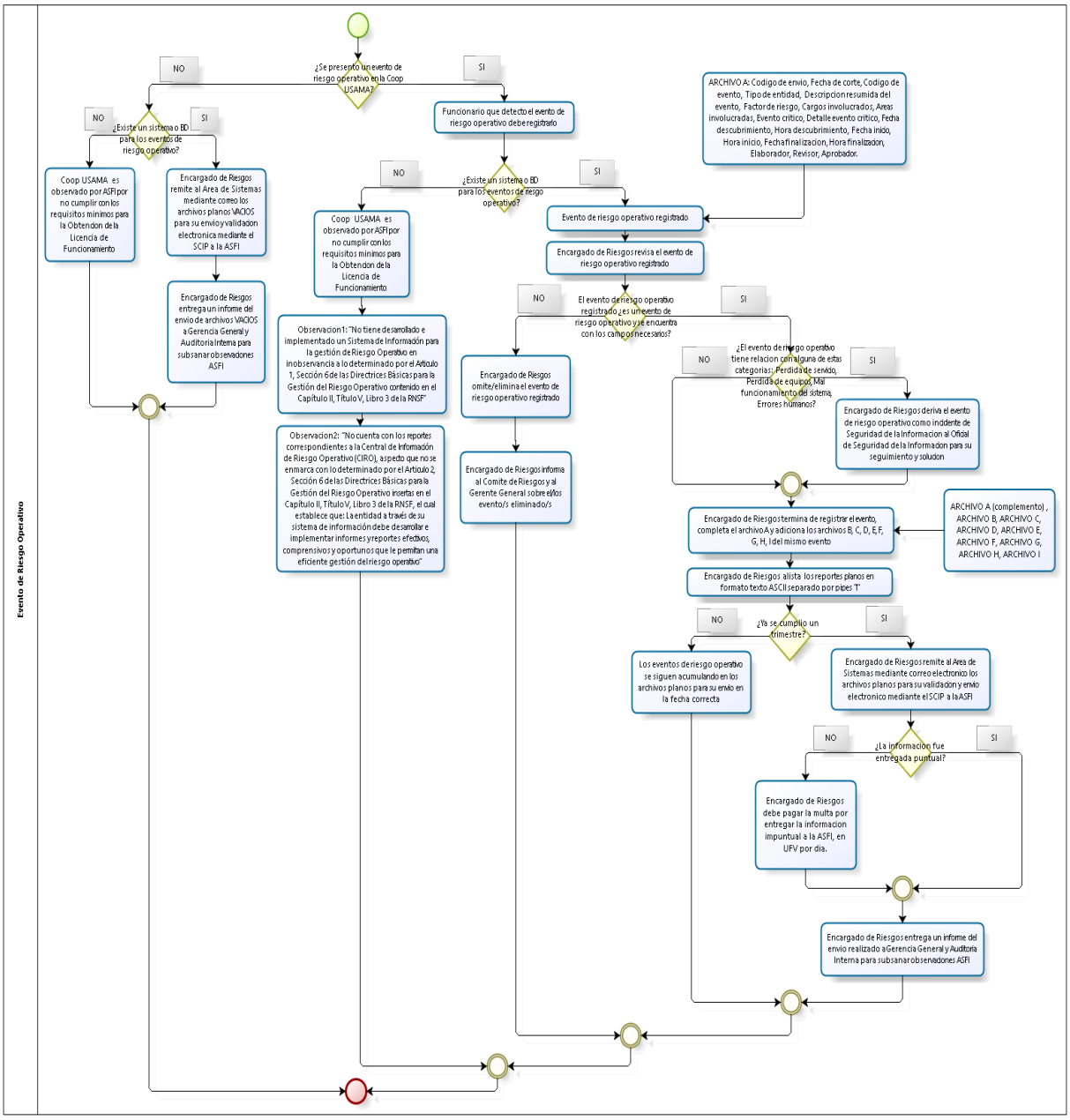
Diagrama de Flujo - Incidente de Seguridad de la Información (Actualmente)



Nota. Diagrama de Flujo – Incidente de Seguridad de la Información, cómo se maneja dentro de la Cooperativa Unión Santiago de Machaca USAMA Ltda. (Actualmente).

Figura 13

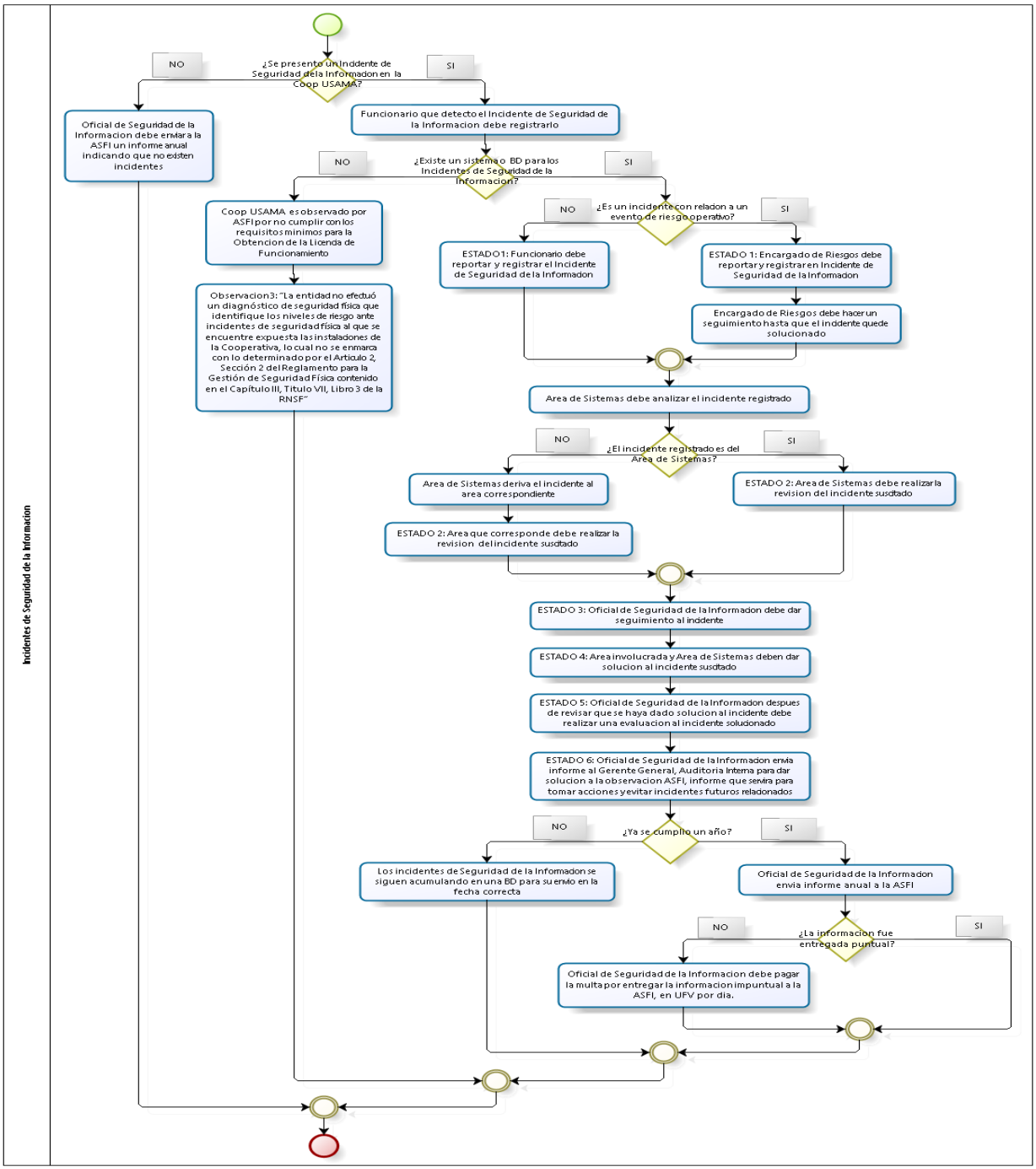
Diagrama de Flujo - Evento de Riesgo Operativo (Propuesta)



Nota. Diagrama de Flujo – Evento de Riesgo Operativo, propuesta de Flujo luego del desarrollo del Sistema para la Cooperativa Unión Santiago de Machaca USAMA Ltda.

Figura 14

Diagrama de Flujo - Incidente de Seguridad de la Información (Propuesta)



Nota. Diagrama de Flujo – Incidente de Seguridad de la Información, propuesta de Flujo luego del desarrollo del Sistema para la Cooperativa Unión Santiago de Machaca USAMA Ltda.

Figura 15

Ubicación de la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

Nota. Ubicación de la Cooperativa USAMA Ltda., generada de Google Maps.

3.2.2. Identificación de actores

La siguiente tabla referente muestra la descripción de los procesos de iteración que tienen los actores con el sistema, especificando las funciones que desempeña cada actor.

Tabla 11

Descripción de actores

ACTOR	DESCRIPCIÓN
OPERADOR (FUNCIONARIO)	El operador son todos los funcionarios que desempeñan profesionalmente un cargo dentro de la Cooperativa USAMA Ltda.
SUPERVISOR (ENCARGADO DE RIESGOS)	El supervisor es el encargado de Identificar, medir, monitorear, controlar, mitigar y divulgar los diferentes tipos de riesgos que enfrenta la Cooperativa en el desempeño de sus actividades, generando políticas,

	procedimientos y estrategias que permitan desvelar y prevenir la mayor o menor exposición al riesgo, de tal manera de informar oportunamente al Gerente General, Consejo de Administración y otras instancias con el fin de que se tomen las acciones que correspondan.
SEGURIDAD (OFICIAL DE SEGURIDAD DE LA INFORMACIÓN)	Seguridad es el encargado de planificar, desarrollar, controlar y gestionar las políticas, procedimientos y acciones con el fin de mejorar la seguridad de la información, manteniendo la confidencialidad, integridad y disponibilidad de la información, ejerciendo responsabilidad sobre las decisiones de seguridad de T.I. y la continuidad del negocio.
SOPORTE (TÉCNICO DE SOPORTE E INFRAESTRUCTURA DE SISTEMAS)	Soporte es el responsable de gestionar las soluciones de la seguridad, manteniendo la confidencialidad, integridad y disponibilidad de la información, en coordinación con el jefe de Sistemas, siguiendo los procedimientos de Seguridad de la Información.
ADMINISTRADOR (JEFE DE SISTEMAS)	El Administrador es el responsable de Administrar el funcionamiento correcto de los Sistemas y realizar mantenimiento del Software y Hardware de la Cooperativa, para garantizar un correcto procesamiento de datos y la obtención de información confiable, integra y oportuna en el marco de las disposiciones legales en vigencia.

Nota. Descripción de los actores que interactuan en el Sistema

3.2.3. Ingeniería de Requerimientos

La ingeniería de requerimientos es una parte importante del desarrollo de software, ya que la misma nos permite identificar las funcionalidades que debería

cumplir el sistema para colmar las expectativas de los usuarios que participan en la misma. Así también se definen las restricciones sobre el sistema.

La descripción clara de los requerimientos a cumplir en el sistema nos evita de ambigüedades, una mejor construcción del software consistente y compacto. La obtención correcta de los requerimientos describe con claridad el comportamiento que tendrá el sistema. Las funciones que debe realizar se clasifican en tres categorías como se detallan en la siguiente tabla:

Tabla 12

Categoría de las funciones

CATEGORÍA DE LA FUNCIÓN	SIGNIFICADO
Evidente	Esta categoría hace referencia a lo que debe realizarse, y el usuario debería saber que se realizó, a través de diferentes tipos de avisos, notificaciones, salidas, informes, etc.
Oculto	Debe realizarse, aunque no necesariamente el usuario debe verificar la misma. Esto puede hacer referencia a procesos como por ejemplo el registro de información, almacenamiento o algún tipo de persistencia de datos, servicios necesarios para la comunicación hardware software, etc.
Opcionales	Su inclusión no repercute significativamente en el costo ni en otras funciones.

Nota. Categorías de las funciones presentes en la Ingeniería de Requerimientos

3.2.3.1. Requerimientos funcionales.

Los requerimientos funcionales nos permiten identificar las funciones primarias propias del sistema las cuales cumplirán los requerimientos de acuerdo al usuario. Una función se describe como un conjunto de entradas, procesos o comportamientos y

salidas. Los requerimientos funcionales, pueden ser: cálculos, detalles técnicos, manipulación de datos como la interactuación con la base de datos o archivos.

Tabla 13

Requerimientos funcionales

REQ.	FUNCIÓN	CATEGORÍA
R1	Acceso únicamente a usuarios verificados y con permisos asignados.	Evidente
R2	Acceso al sistema por roles de usuario (Operador, Supervisor, Seguridad, Soporte, Administrador).	Evidente
R3	Registrar, modificar y eliminar usuarios del sistema.	Evidente
R4	Asignación de permisos de acuerdo al rol y funciones que desempeña.	Evidente
R5	El sistema debe permitir el registro de Eventos de Riesgo Operativo.	Evidente
R6	El sistema debe permitir el registro de Incidentes de Seguridad de la Información.	Evidente
R7	El sistema debe generar reportes individuales.	Evidente
R8	El sistema debe generar reportes por rango de fechas.	Evidente
R9	El sistema debe permitir la búsqueda, modificación y eliminación de los registros (Eventos de Riesgo Operativo).	Evidente

R10	El sistema debe permitir la búsqueda, modificación y eliminación de los registros (Incidentes de Seguridad de la Información).	Evidente
R11	Visualización en un diagrama de barras sobre la cantidad de los Eventos de Riesgo Operativo registrados en el sistema.	Evidente

Nota. Requerimientos funcionales (Elaboración propia)

3.2.3.2. Requerimientos no funcionales.

Son requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, si no a propiedades emergentes como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

Tabla 14

Requerimientos no funcionales

REQ	FUNCIÓN
R1	El sistema debe visualizarse y funcionar correctamente en cualquier navegador como ser: Chrome, Mozilla, Microsoft Edge.
R2	El sistema debe ser instalado en un servidor, con el propósito de estar disponible las 24 horas del día.
R3	Implementar el sistema en un servidor donde se pueda almacenar copias de seguridad de manera sencilla.
R4	El sistema al poseer varios módulos para los diferentes tipos de usuarios, debe tener un módulo de uso administrativo exclusivo, con la finalidad de proteger la integridad de los datos.
R5	El sistema debe asegurar los datos y evitar los accesos no autorizados.

R6	El sistema contará con manuales de usuario adecuados.
R7	La interfaz gráfica debe ser fácil y entendible para el usuario.

Nota. Requerimientos no funcionales (Elaboración propia)

3.3. FASE DE ELABORACIÓN

La fase de elaboración se profundiza en la comprensión de los requisitos del sistema, mediante el modelo de análisis que comprende los casos de uso y con el modelo de diseño se valida la arquitectura.

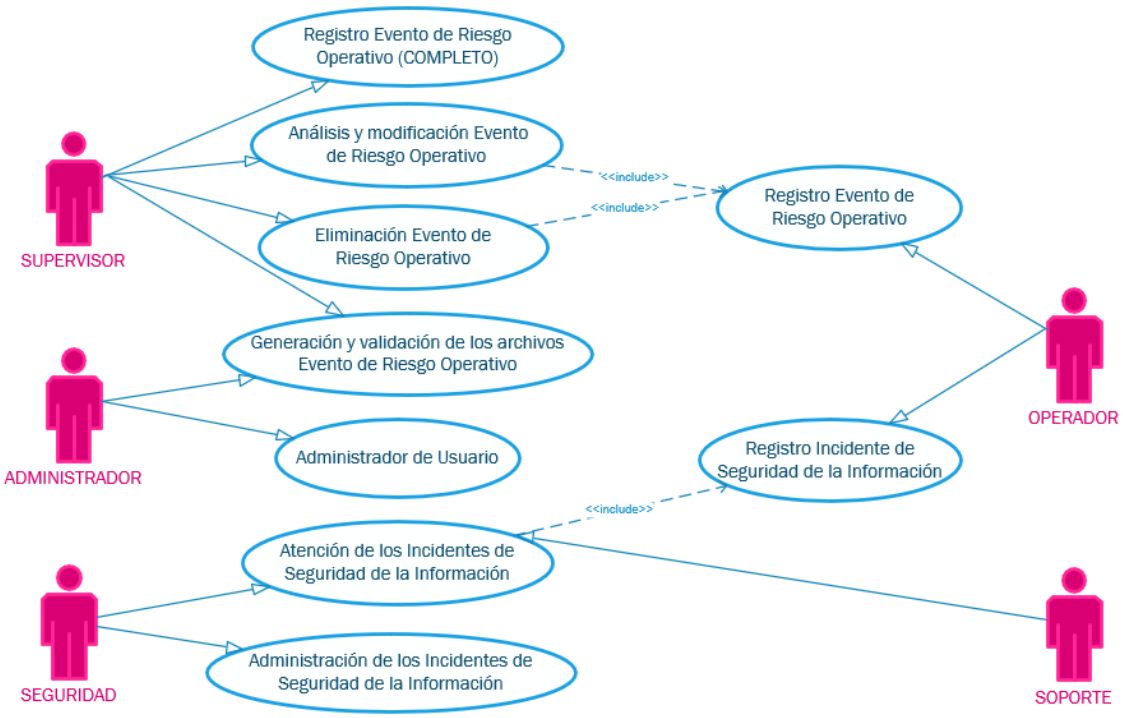
3.3.1. Modelo de caso de uso

Se realiza un análisis de los casos de uso que representa los actores del sistema para luego hacer la descripción de cada uno de ellos. Es así que se tiene los siguientes diagramas de caso de uso.

Diagrama de caso de uso de alto nivel

Figura 16

Casos de Uso del Sistema



Nota. Diagrama de Caso de Uso del Sistema.

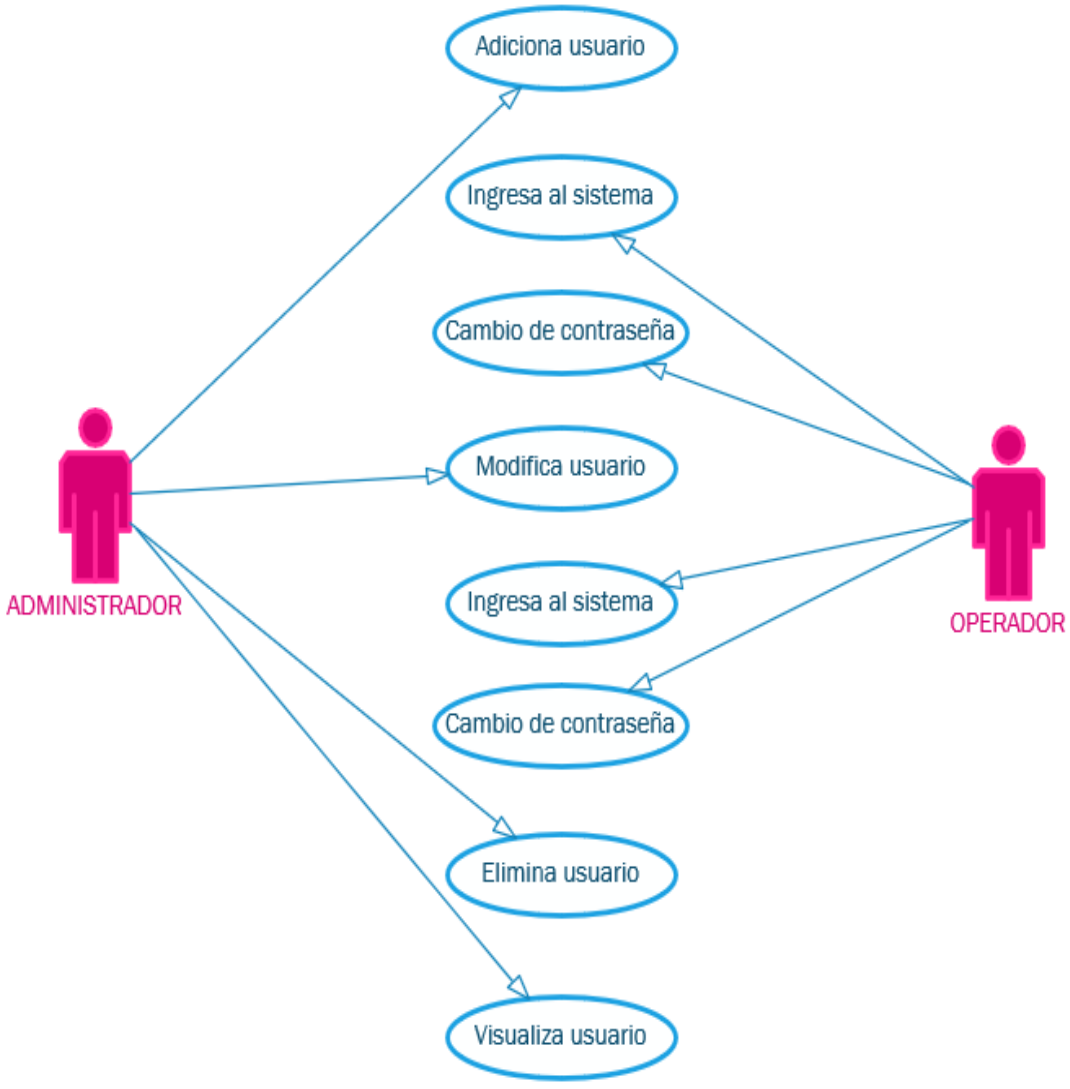
3.3.2. Descripción de caso de uso

A continuación, se mostrarán los casos de uso, de los procesos reflejados en el diagrama de casos de uso del sistema.

Caso de Uso: Administrador de Usuario

Figura 17

Caso de Uso - Administrador de Usuario



Nota. Diagrama de Caso de Uso – Administrador de Usuario (Elaboración propia).

Tabla 15

Descripción de Caso de Uso - Administrador de Usuario

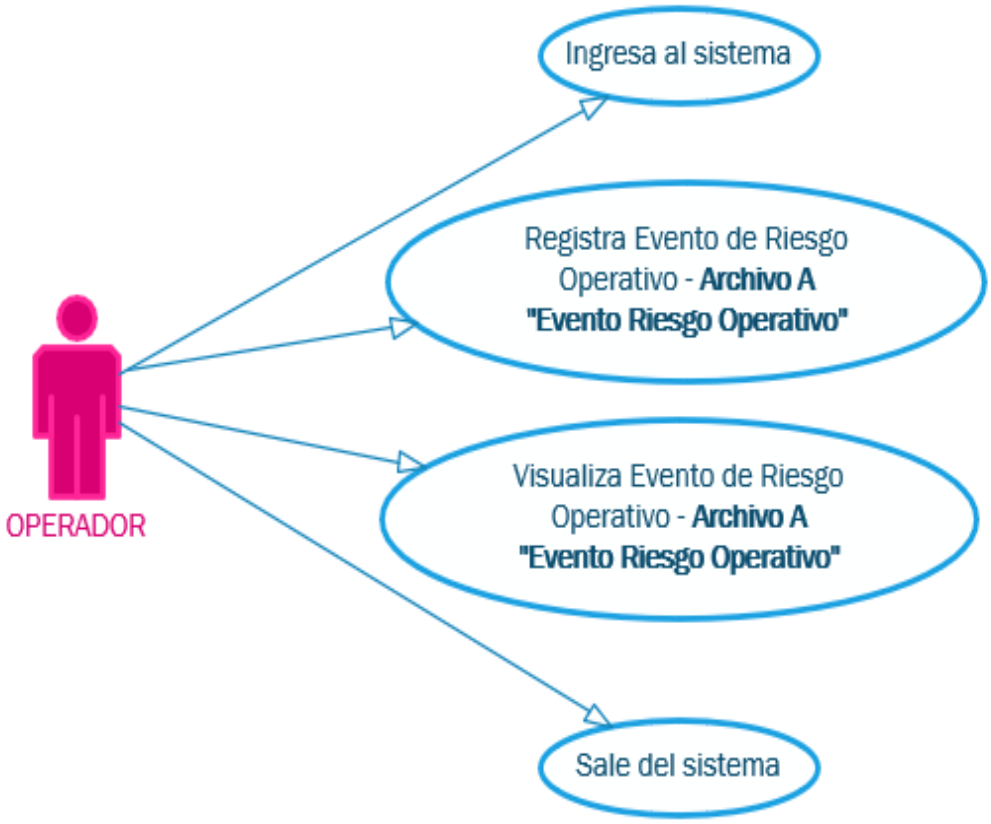
Caso de Uso	**Administrador de Usuario**
Descripción	El administrador realizará el registro, modificación, eliminación y visualización de los usuarios.
Actores	Administrador y Operador.
Precondiciones	El administrador debe estar logueado en el sistema.
Flujo normal	1. El administrador ingresa al sistema. 2. El administrador realiza el registro de usuario. • El operador ingresa al sistema con la contraseña asignada inicialmente. • El operador cambia su contraseña inicial. 3. El administrador realiza la modificación de usuario. • El operador ingresa al sistema con la contraseña asignada después de la modificación. • El operador cambia su contraseña asignada por el administrador. 4. El administrador realiza la eliminación de usuario. 5. El administrador visualiza los cambios realizados (adición, modificación o eliminación). 6. El sistema realiza el almacenamiento de los cambios realizados. 7. El administrador sale del sistema.
Flujo alternativo	Si se realiza algún error de registro, modificación y eliminación el sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso - Administrador de Usuario (Elaboración propia).

Caso de Uso: Registro Evento de Riesgo Operativo

Figura 18

Caso de Uso – Registro Evento de Riesgo Operativo



Nota. Diagrama de Caso de Uso – Registro Evento de Riesgo Operativo (Elaboración propia).

Tabla 16

Descripción de Caso de Uso – Registro de Evento de Riesgo Operativo

Caso de Uso	**Administrador de Usuario**
Descripción	El operador realizará el registro de los Eventos de Riesgo Operativo que se presenten dentro de Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.

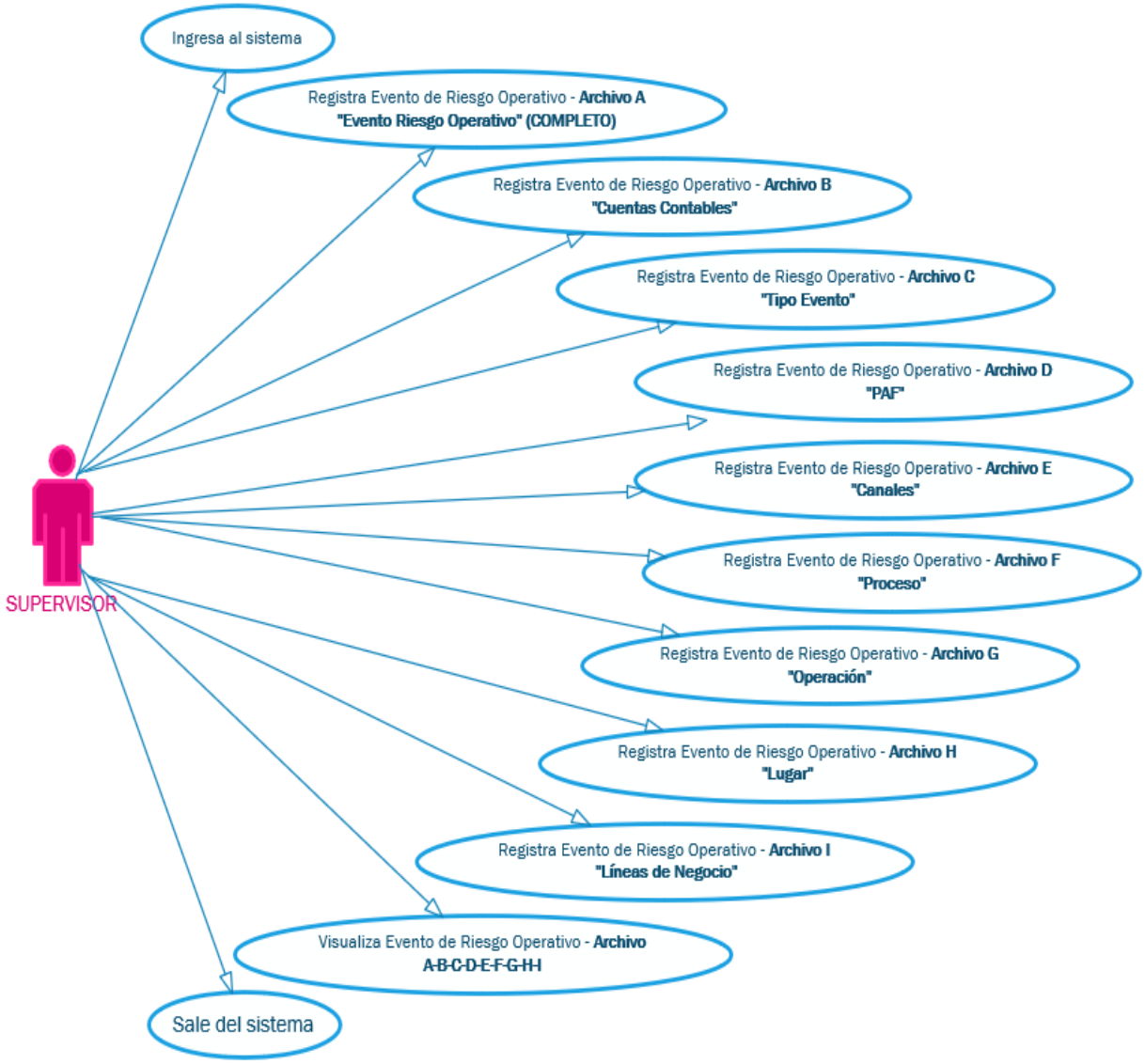
Actores	Operador.
Precondiciones	El Operador debe haber presenciado algún Evento de Riesgo Operativo que sucedió dentro de la Cooperativa USAMA Ltda. El Operador debe estar logueado en el sistema.
Flujo normal	1. El Operador ingresa al sistema. 2. El Operador realiza el registro del Evento de Riesgo Operativo. 3. El Operador corrobora y visualiza el Evento de Riesgo Operativo registrado. 4. El Operador sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Registro Evento de Riesgo Operativo (Elaboración propia).

Caso de Uso: Registro Evento de Riesgo Operativo (COMPLETO)

Figura 19

Caso de Uso – Registro Evento de Riesgo Operativo (COMPLETO)



Nota. Diagrama de Caso de Uso – Registro Evento de Riesgo Operativo "COMPLETO" (Elaboración propia).

Tabla 17

Descripción de Caso de Uso – Registro de Evento de Riesgo Operativo (COMPLETO)

Caso de Uso	**Administrador de Usuario**
Descripción	El Supervisor realizará el registro de los Eventos de Riesgo Operativo que se presenten dentro de Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.
Actores	Supervisor.
Precondiciones	El Supervisor debe haber presenciado algún Evento de Riesgo Operativo que sucedió dentro de la Cooperativa USAMA Ltda. El Supervisor debe estar logueado en el sistema.
Flujo normal	1. El Supervisor ingresa al sistema. 2. El Supervisor realiza el registro del Evento de Riesgo Operativo - Archivo A "Evento Riesgo Operativo" (COMPLETO). 3. El Supervisor registra Evento de Riesgo Operativo - Archivo B "Cuentas Contables". 4. El Supervisor Registra Evento de Riesgo Operativo - Archivo C "Tipo Evento". 5. El Supervisor Registra Evento de Riesgo Operativo - Archivo D "PAF". 6. El Supervisor Registra Evento de Riesgo Operativo - Archivo E "Canales". 7. El Supervisor Registra Evento de Riesgo Operativo - Archivo F "Proceso".

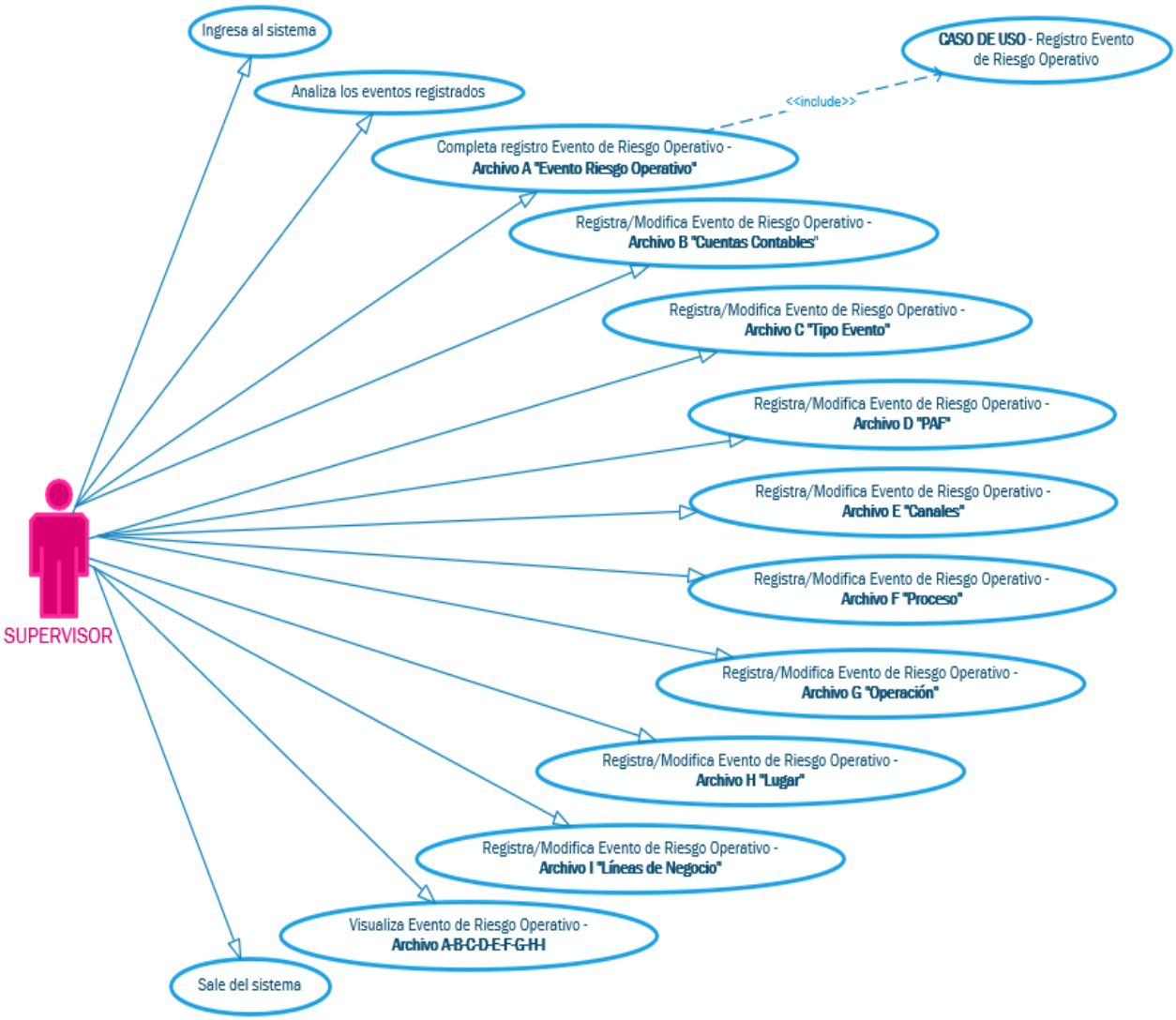
	8. El Supervisor Registra Evento de Riesgo Operativo - Archivo G "Operación". 9. El Supervisor Registra Evento de Riesgo Operativo - Archivo H "Lugar". 10. El Supervisor Registra Evento de Riesgo Operativo - Archivo I "Líneas de Negocio". 11. El Supervisor corrobora, visualiza el Evento de Riesgo Operativo y todo lo relacionado: Archivo A-B-C-D-E-F-G-H-I. 12. El Supervisor sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Registro Evento de Riesgo Operativo "COMPLETO" (Elaboración propia).

Caso de Uso: Análisis y modificación Evento de Riesgo Operativo

Figura 20

Caso de Uso – Análisis y modificación Evento de Riesgo Operativo



Nota. Diagrama de Caso de Uso – Análisis y modificación Evento de Riesgo Operativo (Elaboración propia).

Tabla 18

Descripción de Caso de Uso – Análisis y modificación Evento de Riesgo Operativo

Caso de Uso	**Administrador de Usuario**
Descripción	El Supervisor realizará el registro faltante de los Eventos de Riesgo Operativo que fueron reportados por el Operador, además podrá modificar lo registrado.
Actores	Supervisor.
Precondiciones	Debe existir un Evento de Riesgo Operativo inicial que haya sido registrado por el Operador El Supervisor debe estar logueado en el sistema.
Flujo normal	1. El Supervisor ingresa al sistema. 2. El Supervisor analiza los eventos registrados. 3. El Supervisor completa el registro del Evento de Riesgo Operativo - Archivo A "Evento Riesgo Operativo". 4. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo B "Cuentas Contables". 5. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo C "Tipo Evento". 6. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo D "PAF". 7. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo E "Canales". 8. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo F "Proceso". 9. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo G "Operación".

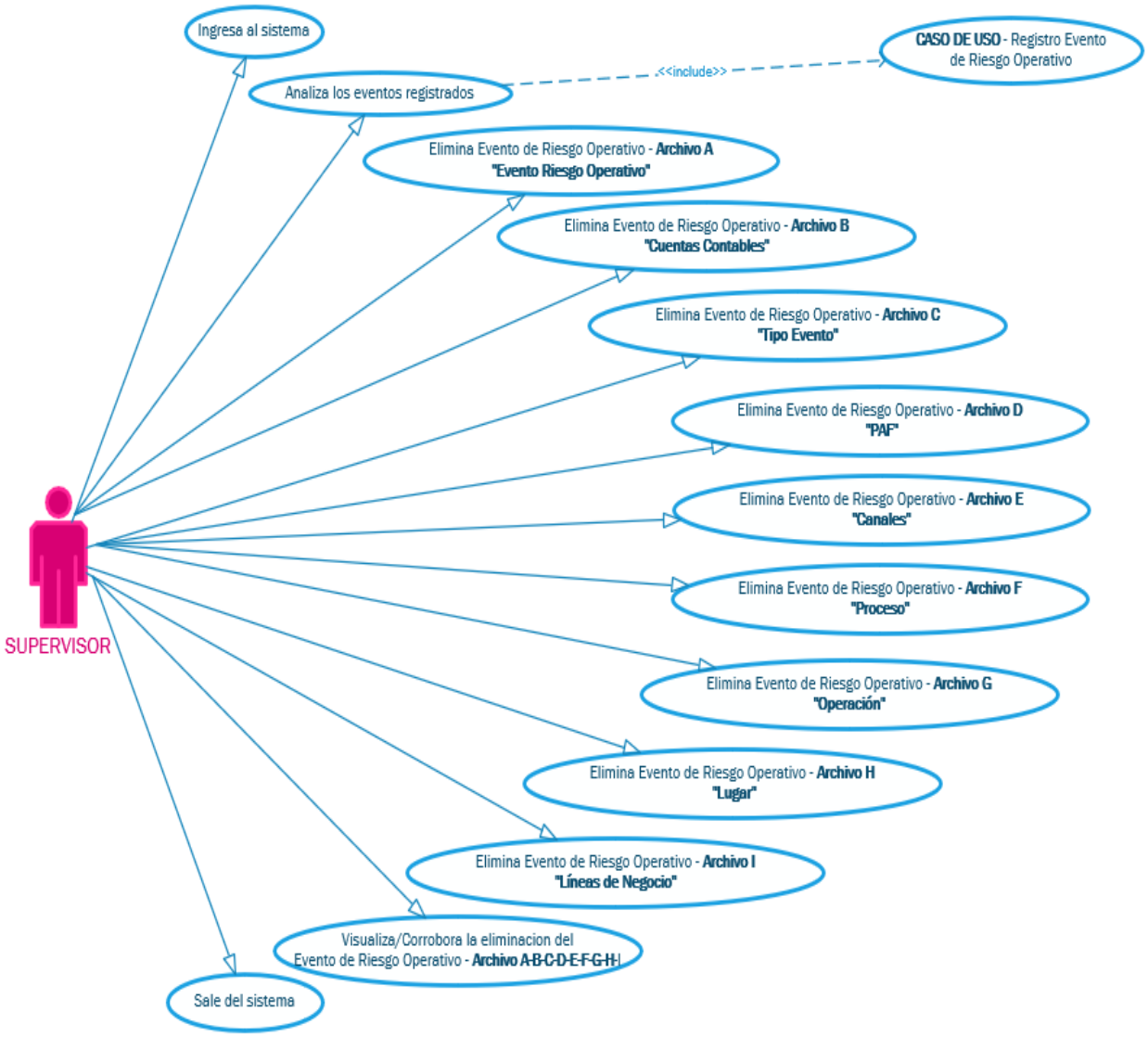
	10. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo H "Lugar". 11. El Supervisor registra y modifica el Evento de Riesgo Operativo - Archivo I "Líneas de Negocio". 12. El Supervisor corrobora, visualiza el Evento de Riesgo Operativo y todo lo relacionado: Archivo A-B-C-D-E-F-G-H-I. 13. El Supervisor sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Análisis y modificación Evento de Riesgo Operativo (Elaboración propia).

Caso de Uso: Eliminación Evento de Riesgo Operativo

Figura 21

Caso de Uso – Eliminación Evento de Riesgo Operativo



Nota. Diagrama de Caso de Uso – Eliminación Evento de Riesgo Operativo (Elaboración propia).

Tabla 19

Descripción de Caso de Uso – Eliminación Evento de Riesgo Operativo

Caso de Uso	**Administrador de Usuario**
Descripción	El Supervisor realizará un análisis de todo lo registrado por el Operador y si no corresponde realizará la eliminación de lo registrado y todo lo relacionado con el Evento de Riesgo Operativo.
Actores	Supervisor.
Precondiciones	Debe existir un Evento de Riesgo Operativo inicial que haya sido registrado por el Operador. El Supervisor debe estar logueado en el sistema.
Flujo normal	1. El Supervisor ingresa al sistema. 2. El Supervisor analiza los eventos registrados. 3. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo A "Evento Riesgo Operativo". 4. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo B "Cuentas Contables". 5. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo C "Tipo Evento". 6. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo D "PAF". 7. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo E "Canales". 8. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo F "Proceso". 9. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo G "Operación".

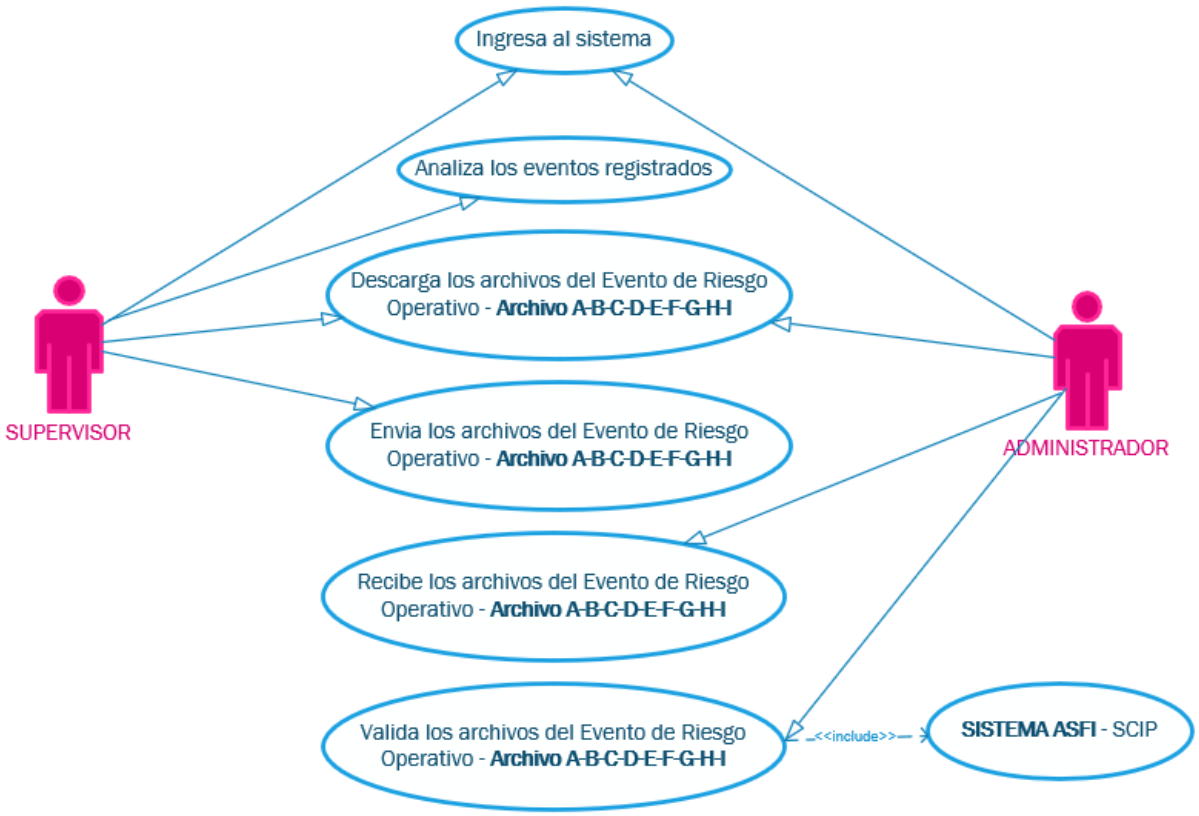
	10. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo H "Lugar". 11. El Supervisor elimina el registro del Evento de Riesgo Operativo - Archivo I "Líneas de Negocio". 12. El Supervisor visualiza/corrobora la eliminación del Evento de Riesgo Operativo y todo lo relacionado: Archivo A-B-C-D-E-F-G-H-I. 13. El Supervisor sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Eliminación Evento de Riesgo Operativo (Elaboración propia).

Caso de Uso: Generación y validación de los archivos Evento de Riesgo Operativo

Figura 22

Caso de Uso – Generación y validación de los archivos Evento de Riesgo Operativo



Nota. Diagrama de Caso de Uso – Generación y validación de los archivos Evento de Riesgo Operativo (Elaboración propia).

Tabla 20

Descripción de Caso de Uso – Generación y validación de los archivos Evento de Riesgo Operativo

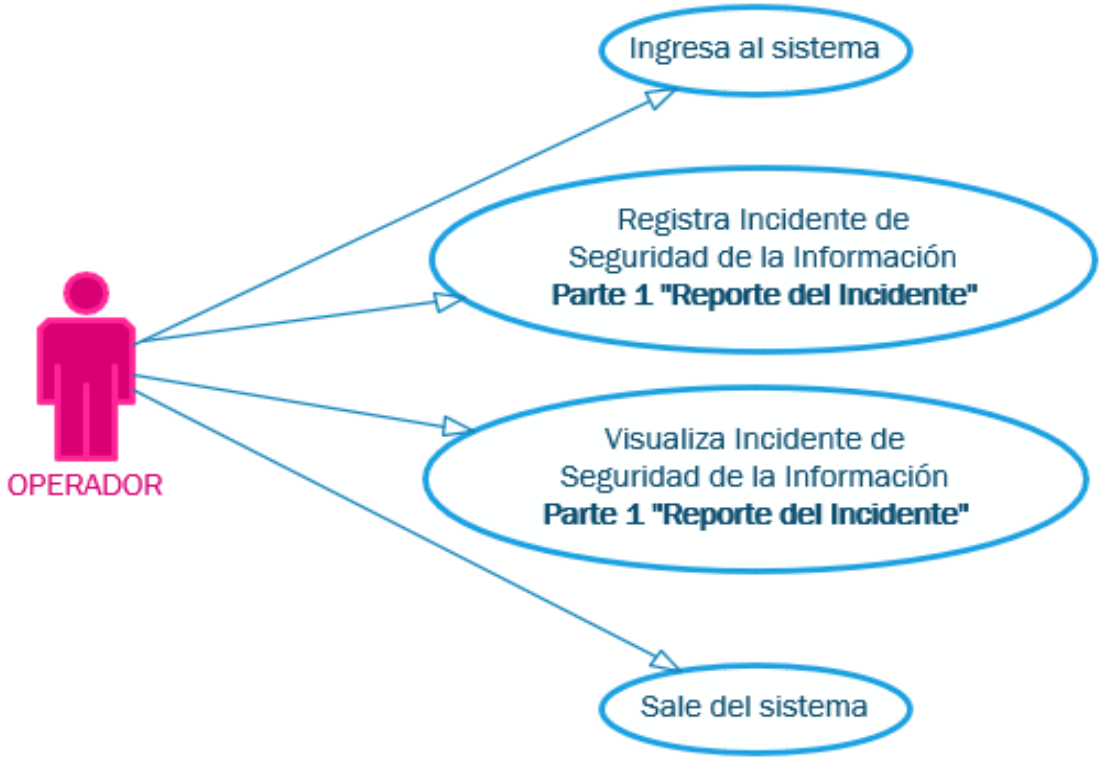
Caso de Uso	**Administrador de Usuario**
Descripción	El Supervisor realizará un análisis de todo lo registrado por el Operador, si corresponde descarga los archivos y envía al Administrador.
Actores	Supervisor, Administrador
Precondiciones	Debe existir un registro de Evento de Riesgo Operativo acumulado. El Supervisor y Administrador deben estar logueado en el sistema.
Flujo normal	1. El Supervisor y Administrador ingresan al sistema. 2. El Supervisor analiza los eventos registrados. 3. El Supervisor descarga los archivos del Evento de Riesgo Operativo - Archivo A-B-C-D-E-F-G-H-I. 4. El Supervisor envía al Administrador los archivos del Evento de Riesgo Operativo - Archivo A-B-C-D-E-F-G-H-I. 5. El Administrador recibe los archivos del Evento de Riesgo Operativo - Archivo A-B-C-D-E-F-G-H-I. 6. El Administrador valida los archivos del Evento de Riesgo Operativo - Archivo A-B-C-D-E-F-G-H-I en el Sistema de Captura de Información Periódica SCIP.
Flujo alternativo	Si no se sigue el flujo, no se realizará a tiempo el envió de la información trimestral.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Generación y validación de los archivos Evento de Riesgo Operativo (Elaboración propia).

Caso de Uso: Registro Evento de Riesgo Operativo

Figura 23

Caso de Uso – Registro Incidente de Seguridad de la Información



Nota. Diagrama de Caso de Uso – Registro Incidente de Seguridad de la Información (Elaboración propia)

.

Tabla 21

Descripción de Caso de Uso – Registro de Incidente de Seguridad de la Información

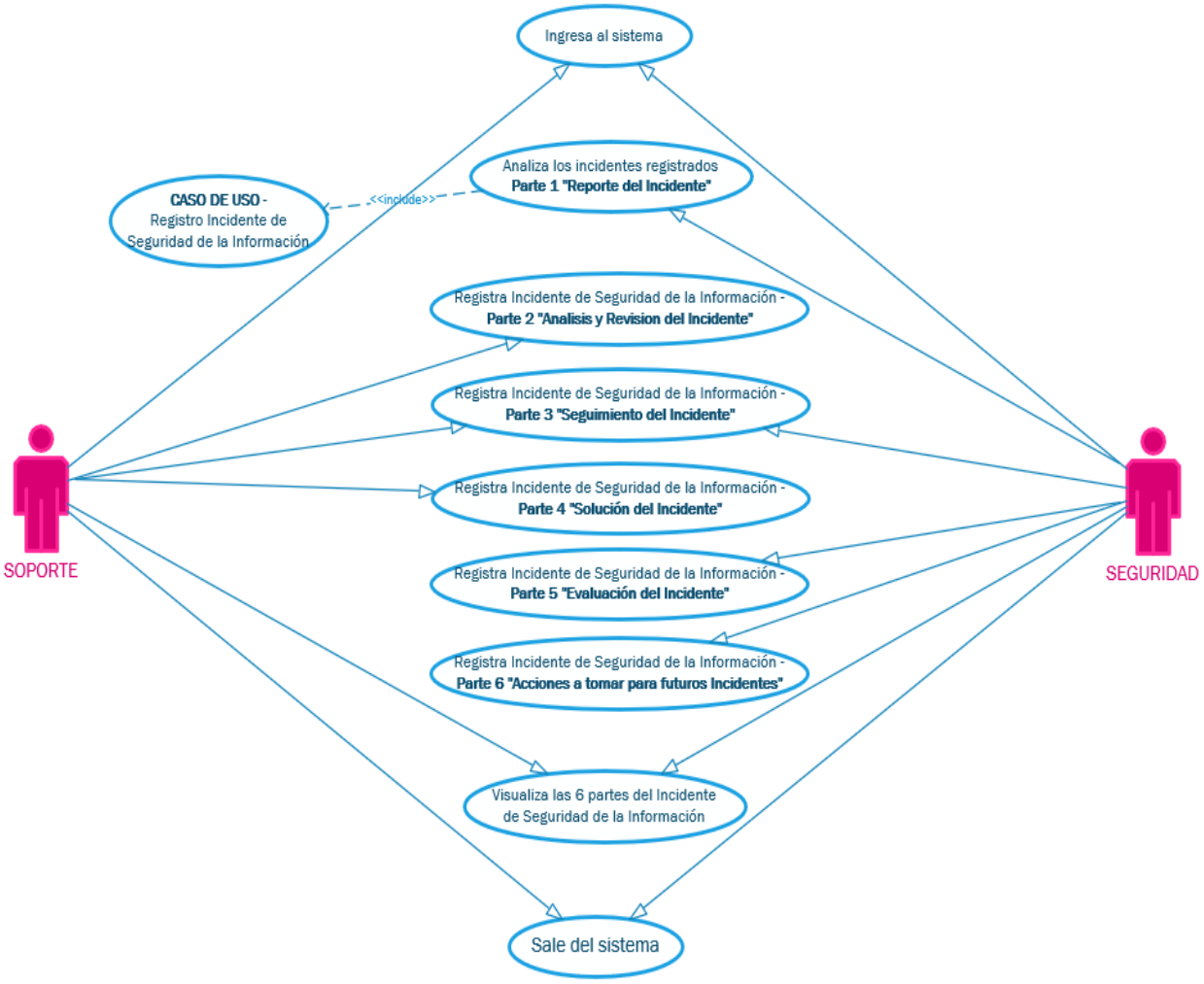
Caso de Uso	**Administrador de Usuario**
Descripción	El operador realizará el registro de los Incidentes de Seguridad de la Información que se presenten dentro de Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.
Actores	Operador.
Precondiciones	El Operador debe haber presenciado algún Incidente de Seguridad de la Información que sucedió dentro de la Cooperativa USAMA Ltda. El Operador debe estar logueado en el sistema.
Flujo normal	1. El Operador ingresa al sistema. 2. El Operador realiza el registro del Incidente de Seguridad de la Información **Parte 1 "Reporte del Incidente".** 3. El Operador corrobora y visualiza el Incidente de Seguridad de la Información **Parte 1 "Reporte del Incidente"** que fue registrado. 4. El Operador sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Registro Incidente de Seguridad de la Información (Elaboración propia).

Caso de Uso: Atención a los Incidentes de Seguridad de la Información

Figura 24

Caso de Uso – Atención de los Incidentes de Seguridad de la Información



Nota. Diagrama de Caso de Uso – Atención de los Incidentes de Seguridad de la Información (Elaboración propia).

Tabla 22

Descripción de Caso de Uso – Atención de los Incidentes de Seguridad de la Información

Caso de Uso	**Administrador de Usuario**
Descripción	El rol Seguridad realizará un análisis de todo lo registrado por el Operador sobre Incidentes de Seguridad de la Información, para que se le de atención por parte del Soporte y el de Seguridad.
Actores	Seguridad y Soporte.
Precondiciones	Debe existir un Incidente de Seguridad de la Información inicial que haya sido registrado por el Operador. Seguridad y el Soporte deben estar logueado en el sistema.
Flujo normal	1. Seguridad y el Soporte ingresan al sistema. 2. Seguridad analiza los incidentes registrados **Parte 1 “Reporte del Incidente”.** 3. El Soporte registra el Incidente de Seguridad de la Información **Parte 2 "Análisis y Revisión del Incidente".** 4. Seguridad y Soporte registran el Incidente de Seguridad de la Información **Parte 3 "Seguimiento del Incidente".** 5. Soporte registra el Incidente de Seguridad de la Información **Parte 4 "Solución del Incidente".** 6. Seguridad registra el Incidente de Seguridad de la Información **Parte 5 "Evaluación del Incidente".**

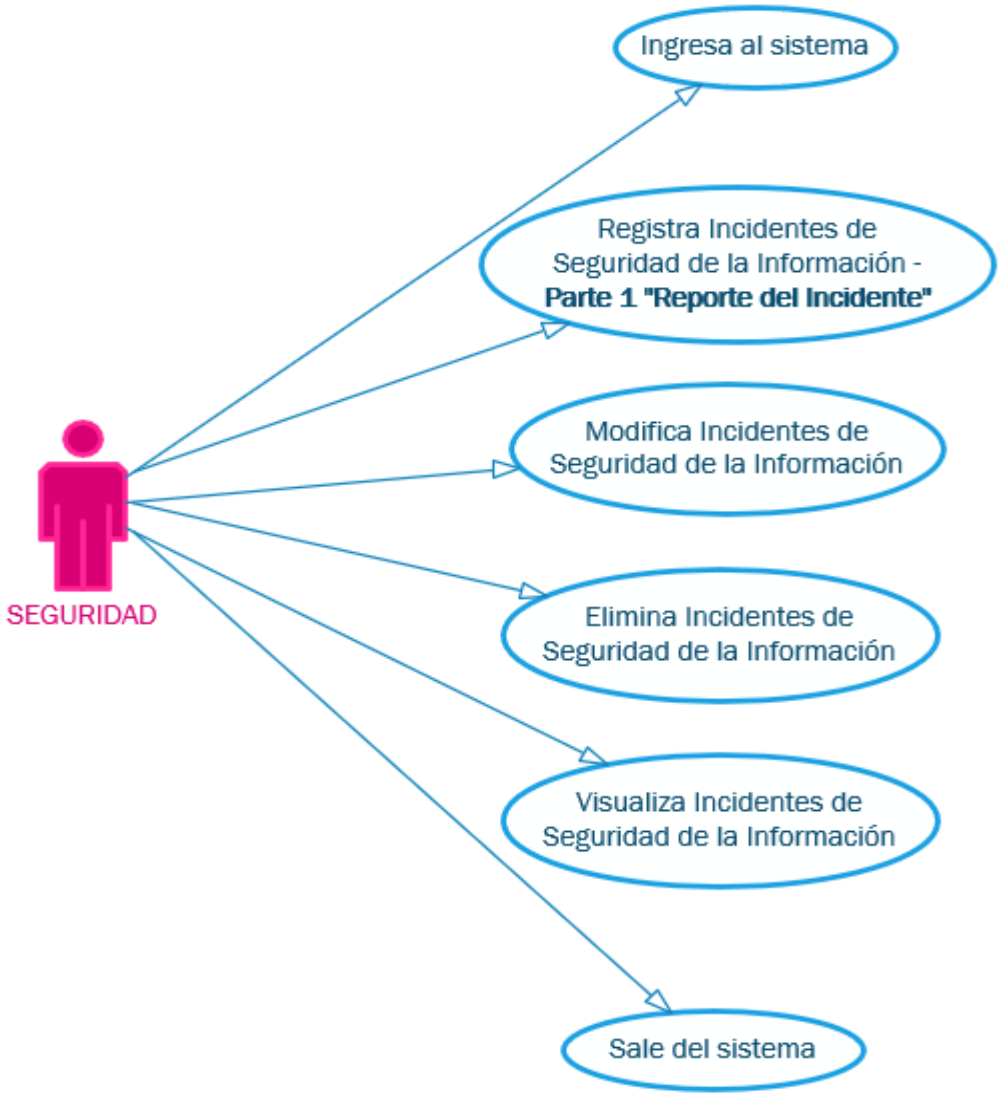
	7. Seguridad registra el Incidente de Seguridad de la Información **Parte 6 "Acciones a tomar para futuros Incidentes".** 8. Seguridad y Soporte visualizan/corroboran las 6 partes del Incidente de Seguridad de la Información y todo lo relacionado. 9. Seguridad y Soporte salen del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

Nota. Descripción de Caso de Uso – Atención de los Incidentes de Seguridad de la Información (Elaboración propia).

Caso de Uso: Administración de los Incidentes de Seguridad de la Información

Figura 25

Caso de Uso – Administración de los Incidentes de Seguridad de la Información



Nota. Diagrama de Caso de Uso – Administración de los Incidentes de Seguridad de la Información (Elaboración propia).

Tabla 23

Descripción de Caso de Uso – Administración de los Incidentes de Seguridad de la Información

Caso de Uso	**Administrador de Usuario**
Descripción	El rol Seguridad administrará los Incidentes de Seguridad de la Información.
Actores	Seguridad.
Precondiciones	Debe existir un Incidente de Seguridad de la Información registrado. Seguridad debe estar logueado en el sistema.
Flujo normal	1. Seguridad ingresa al sistema. 2. Seguridad registra Incidentes de Seguridad de la Información **Parte 1 "Reporte del Incidente".** 3. Seguridad modifica Incidentes de Seguridad de la Información. 4. Seguridad elimina Incidentes de Seguridad de la Información. 5. Seguridad visualizan/corroboran los cambios efectuados sobre Incidentes de Seguridad de la Información y todo lo relacionado. 6. Seguridad sale del sistema.
Flujo alternativo	Si se comete algún error en el registro dentro del sistema, el mismo sistema informará dicho error.
Post Condiciones	El sistema guardará todos los cambios.

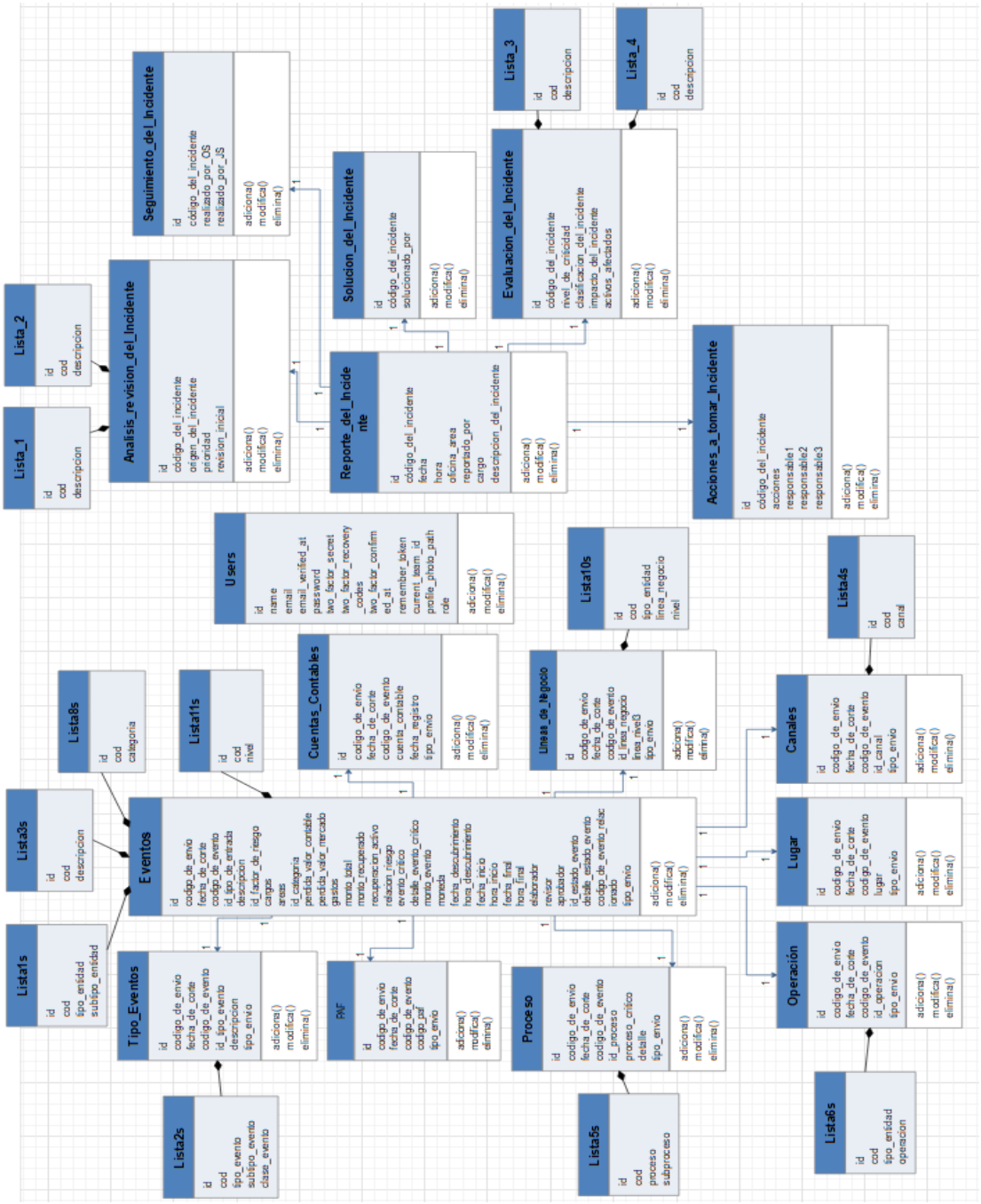
Nota. Descripción de Caso de Uso – Administración de los Incidentes de Seguridad de la Información (Elaboración propia).

3.3.3. Modelo de Clases

En el modelo de clases describimos las clases identificadas, atributos y métodos.

Figura 26

Diagrama de Clases



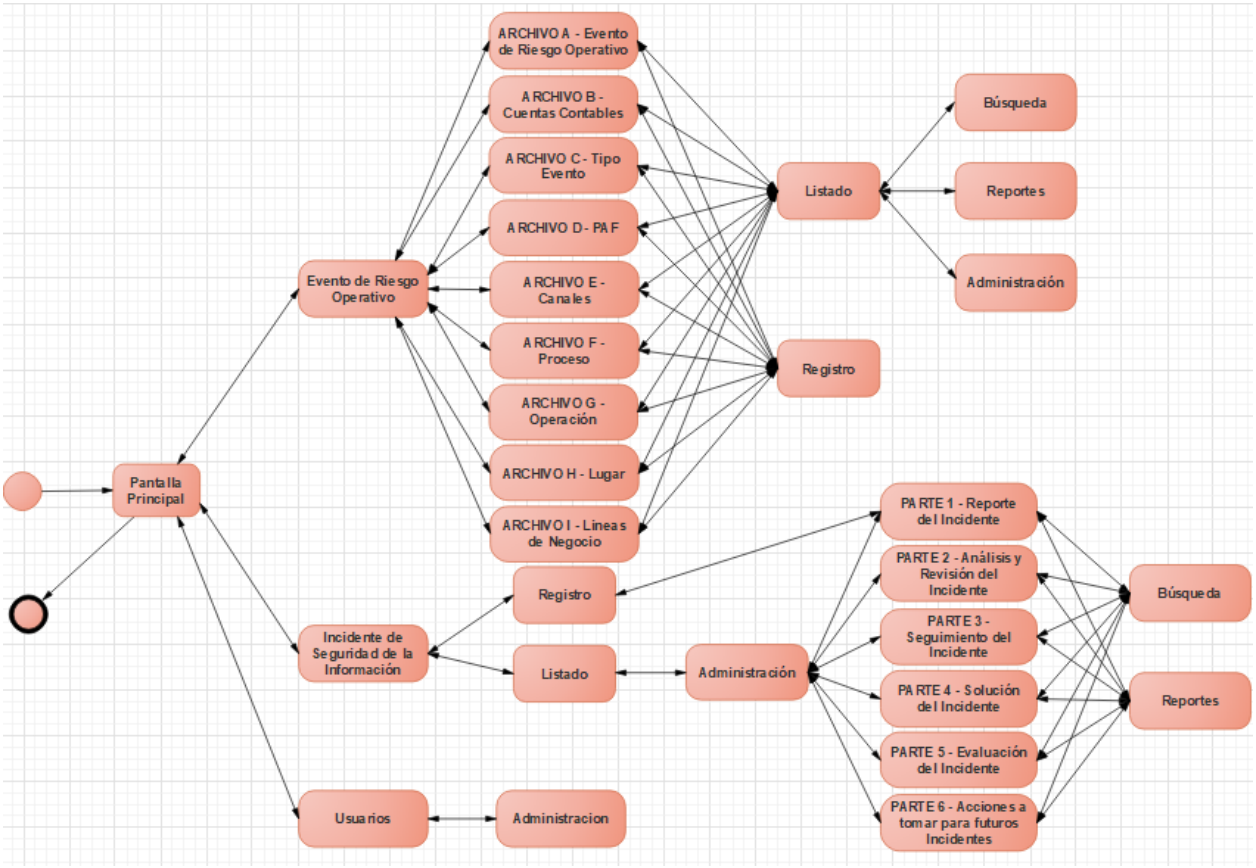
Nota. Diagrama de Clases (Elaboración propia).

3.3.4. Modelo Navegacional

En la siguiente figura observaremos el espacio navegacional que serán visitados mediante la navegación de usuarios.

Figura 27

Modelo Navegacional



Nota. Modelo Navegacional (Elaboración propia).

3.3.5. Modelo de diseño

Se realiza a continuación el modelo entidad relación.

Figura 28

Modelo Entidad Relación

Nota. Modelo Entidad - Relación (Elaboración propia).

3.3.6. Modelo de presentación

El modelo de presentación muestra el cómo se verá el sistema, con una descripción secuencial.

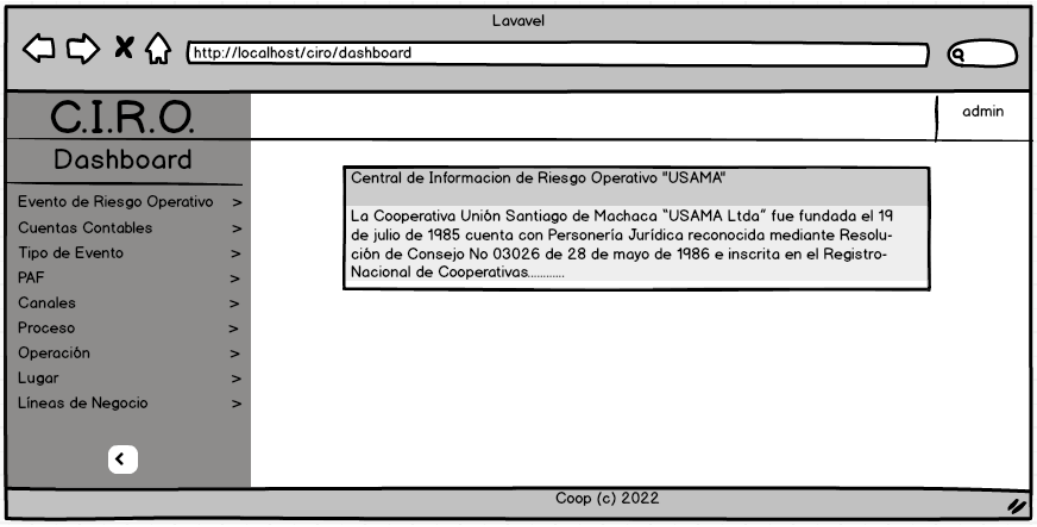
Figura 29

Modelo de presentación - Inicio de sesión al sistema

Nota. Interfaz de ingreso al Sistema, donde el Usuario ingresa sus credenciales de acceso (Elaboración propia).

Figura 30

Modelo de presentación - Bienvenida Evento de Riesgo Operativo



Nota. Interfaz de Bienvenida "Evento de Riesgo Operativo" (Elaboración propia).

Figura 31

Modelo de presentación - Listado Eventos de Riesgo Operativo

Nota. Interfaz de Listado Eventos de Riesgo Operativo (Elaboración propia).

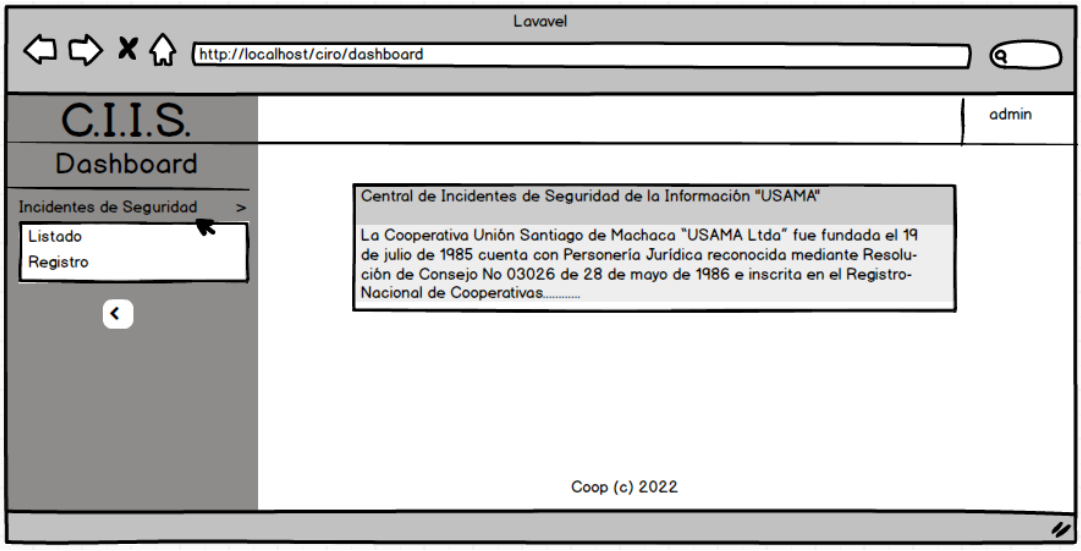
Figura 32

Modelo de presentación - Registro Eventos de Riesgo Operativo

Nota. Interfaz de Registro Eventos de Riesgo Operativo (Elaboración propia).

Figura 33

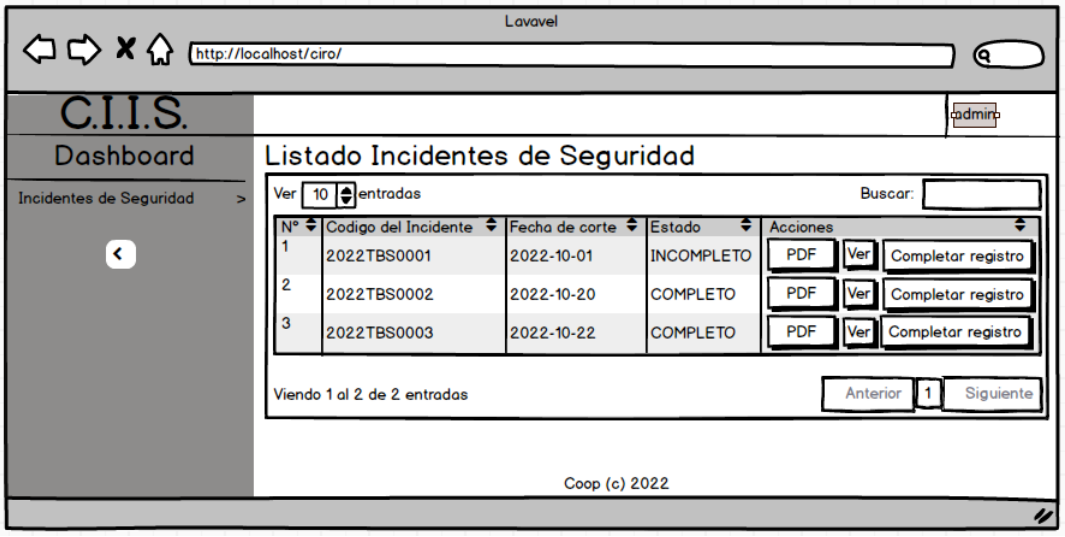
Modelo de presentación - Bienvenida Incidentes de Seguridad de la Información



Nota. Interfaz de Bienvenida "Incidente de Seguridad de la Información" (Elaboración propia).

Figura 34

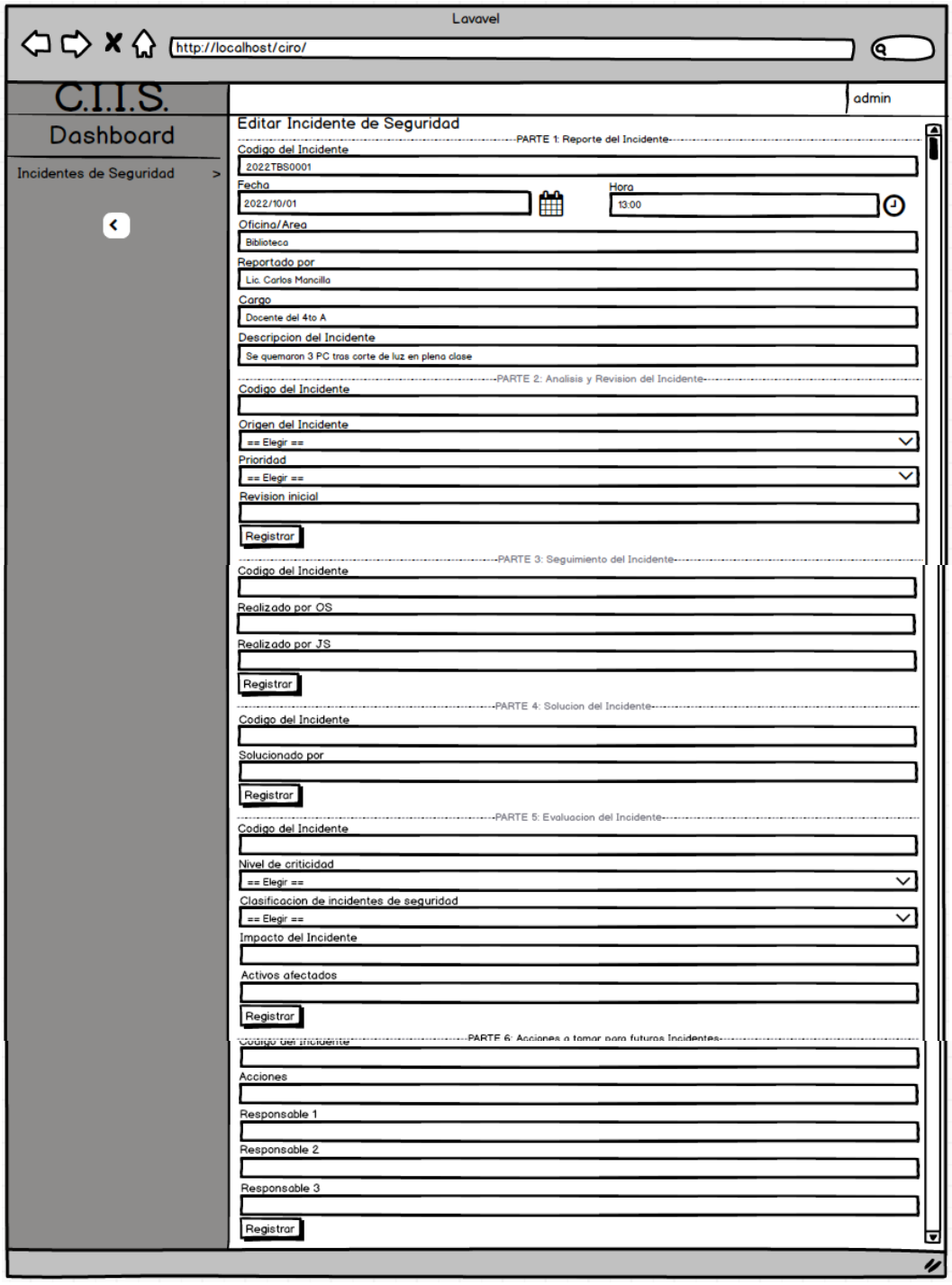
Modelo de presentación - Listado Incidentes de Seguridad de la Información



Nota. Interfaz de Listado Incidentes de Seguridad de la Información (Elaboración propia).

Figura 35

Modelo de presentación – Registro Incidentes de Seguridad de la Información



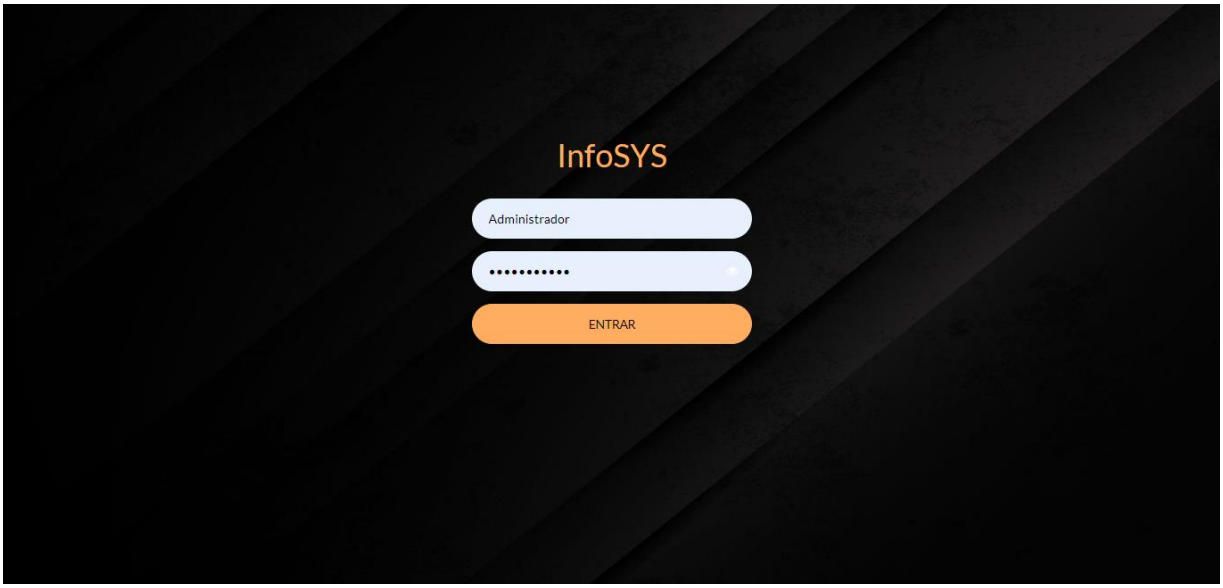
Nota. Interfaz de Registro Incidentes de Seguridad de la Información (Elaboración propia).

3.4. FASE DE CONSTRUCCIÓN

3.4.1. Diseño de interfaz

Figura 36

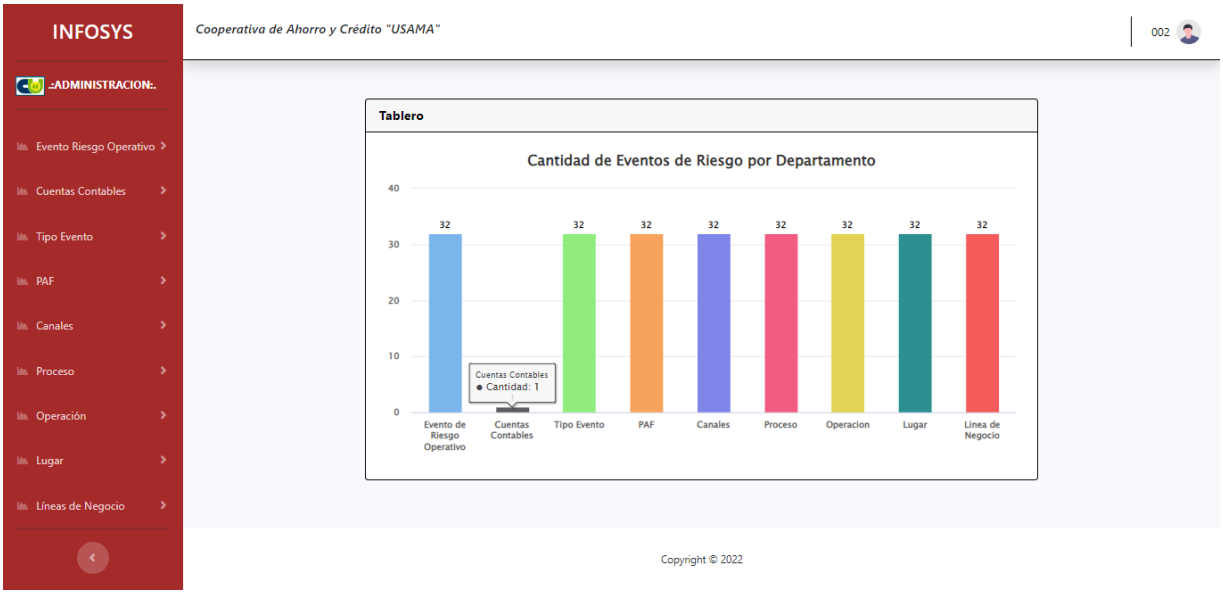
Autentificación al Sistema



Nota. Inicio de sesión – Autentificación al sistema (Elaboración propia).

Figura 37

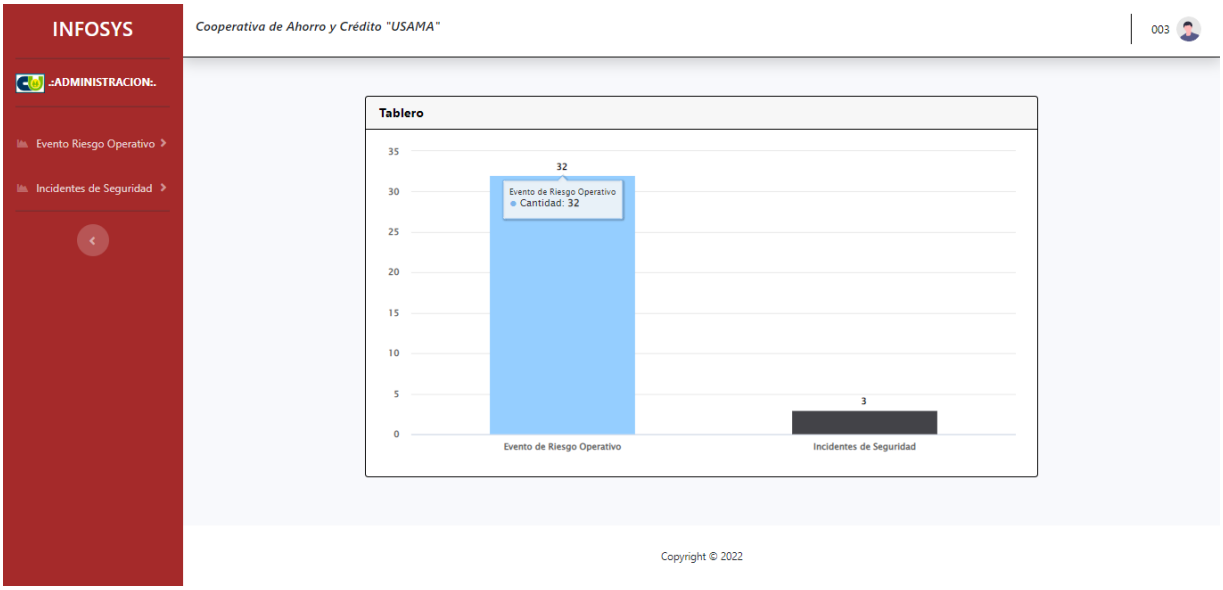
Pantalla de Bienvenida al Sistema (Encargado de Riesgos)



Nota. Pantalla de Bienvenida al Sistema (Encargado de Riesgos), luego de la Autentificación al sistema (Elaboración propia).

Figura 38

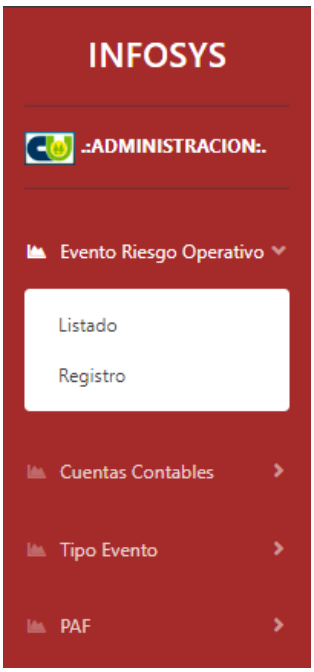
Pantalla de Bienvenida al Sistema (funcionarios)



Nota. Pantalla de Bienvenida al Sistema (funcionarios), luego de la Autentificación al sistema (Elaboración propia).

Figura 39

Panel Lateral de Administración del Sistema



Nota. Panel Lateral de Administración del Sistema "Eventos de Riesgo Operativo" (Elaboración propia).

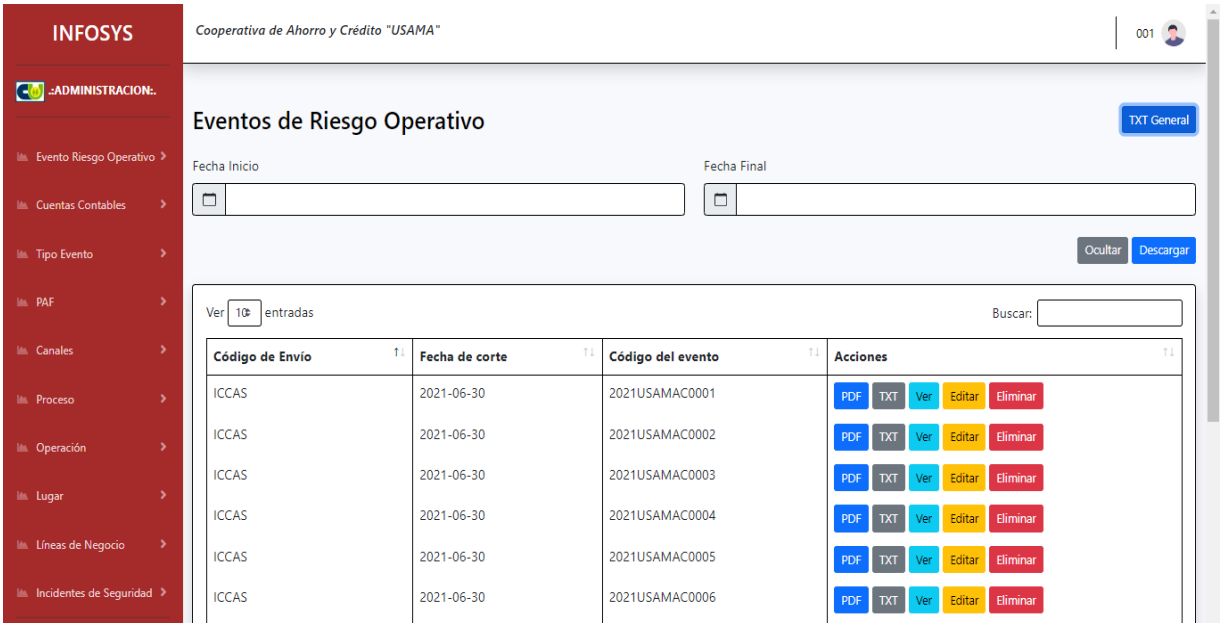
Figura 40

Pantalla de Registro "Eventos de Riesgo Operativo"

Nota. Pantalla de Registro "Eventos de Riesgo Operativo" (Elaboración propia).

Figura 41

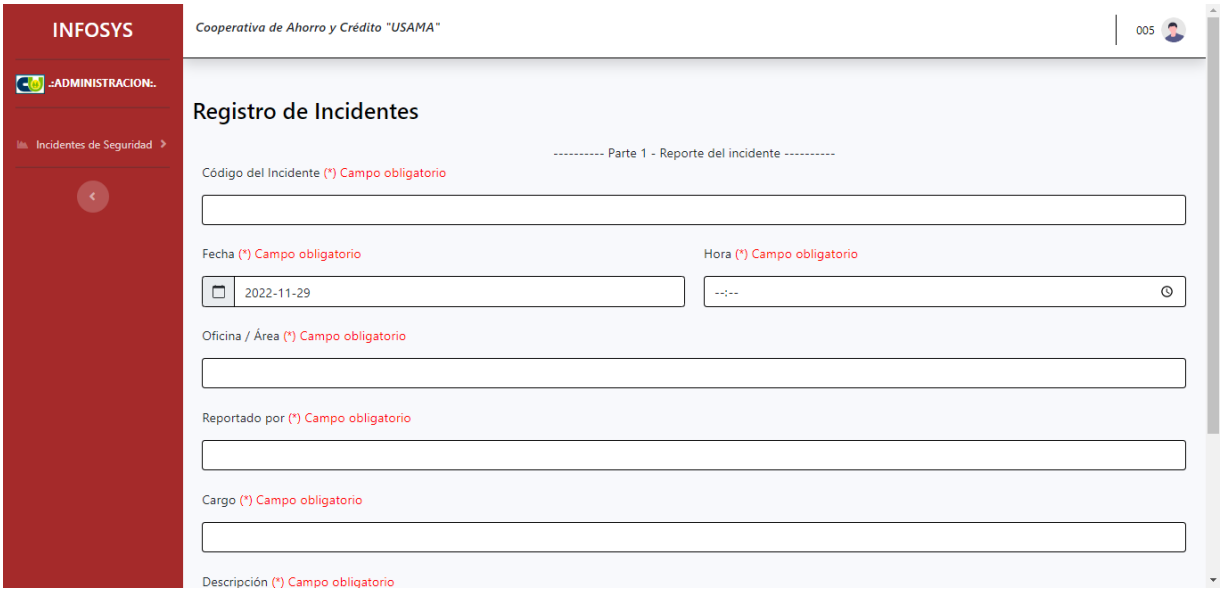
Pantalla Administración - Listado Eventos de Riesgo Operativo



Nota. Pantalla Administración - Listado Eventos de Riesgo Operativo (Elaboración propia).

Figura 42

Pantalla de Registro "Incidentes de Seguridad de la Información"



Nota. Pantalla de Registro "Incidentes de Seguridad de la Información" (Elaboración propia).

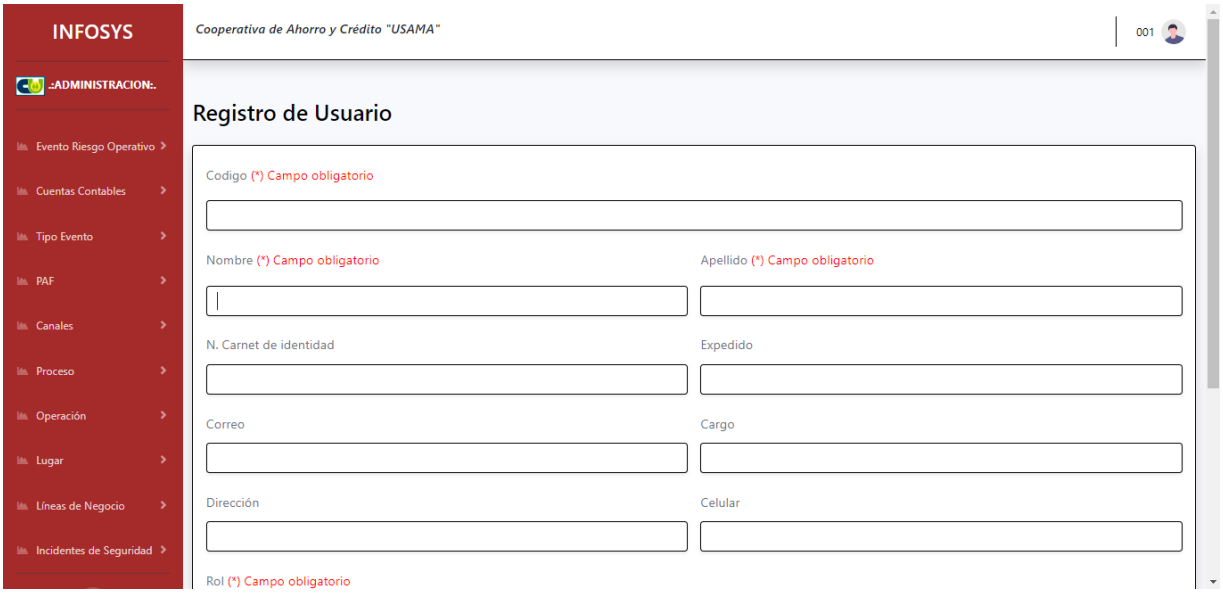
Figura 43

Pantalla Administración - Listado Incidentes de Seguridad de la Información

Nota. Pantalla Administración - Listado Incidentes de Seguridad de la Información (Elaboración propia).

Figura 44

Registro de Usuarios nuevos al Sistema



Nota. Registro de usuarios nuevos al sistema (Elaboración propia).

3.5. FASE DE TRANSICIÓN

3.5.1. Prueba de Software

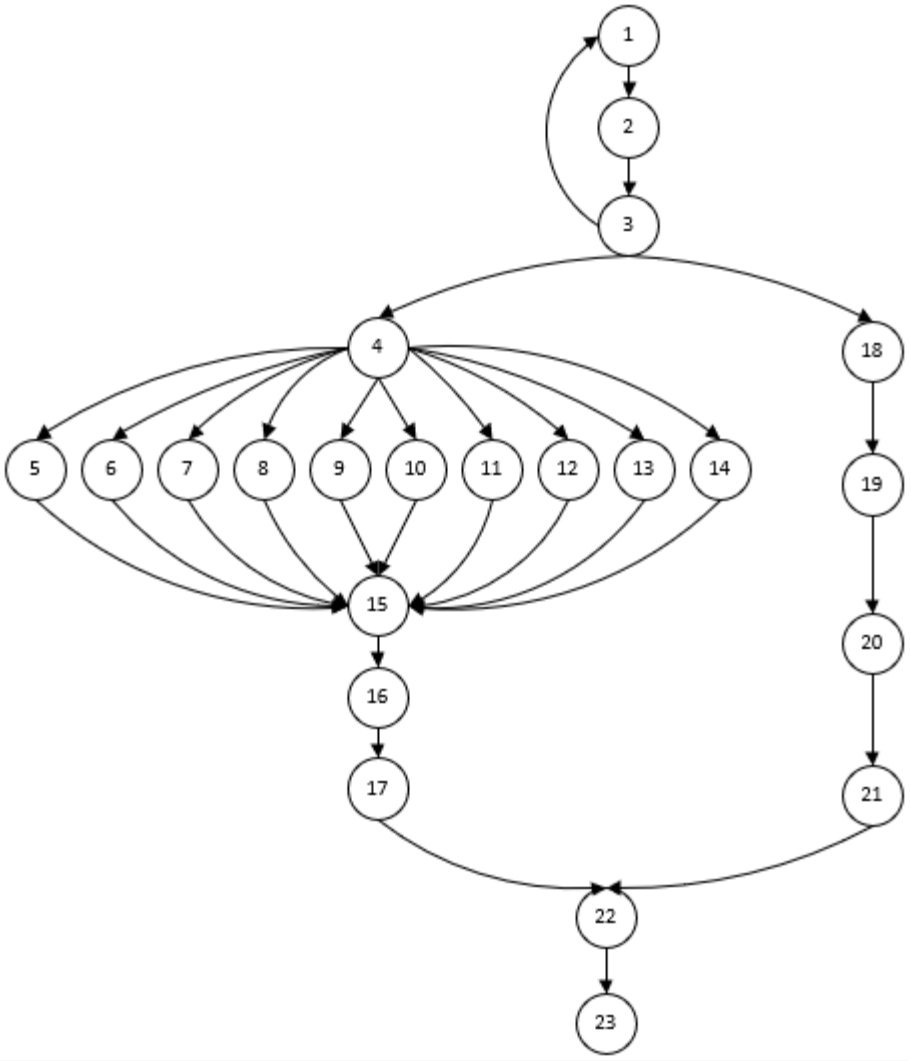
En esta fase se realiza las pruebas de validación para determinar el correcto funcionamiento del sistema.

3.5.1.1. Prueba de Caja blanca.

Permite obtener una medida de complejidad de un diseño procedimental, y utilizar esta medida para definir una serie de caminos básicos de ejecución, garantizando que cada camino se ejecute al menos una vez.

Figura 45

Grafo del Sistema



Nota. Grafo del Sistema (Elaboración propia).

Donde:

1. Inicio del sistema
2. Usuario y contraseña
3. Validar usuario y contraseña
4. Menú principal -Selecciona opción
5. Evento Riesgo Operativo
6. Cuentas Contables
7. Tipo Evento

8. PAF
9. Canales
10. Proceso
11. Operación
12. Lugar
13. Líneas de Negocio
14. Incidentes de Seguridad
15. Registrar Eventos de Riesgo/Incidentes de Seguridad
16. Mostrar Eventos de Riesgo/Incidentes de Seguridad
17. Reportes Eventos de Riesgo/Incidentes de Seguridad
18. Menú administrador
19. Registrar Usuarios
20. Mostrar Usuarios
21. Reporte Usuarios
22. Fin ciclo sistema
23. Fin sistema

Examinamos el grafo creado a partir de las características del sistema, se procede a determinar la complejidad Ciclomática del grafo mediante:

$$V(G)=A-N+2$$

Donde:

A = número de aristas

N = número de nodos

$$V(G)=32-23+2=11$$

Por tanto, la complejidad ciclomática es: V(G) = 12, esto significa que existe 8 caminos independientes.

Camino 1: 1, 2, 3, 4, 5, 15, 16, 17, 22, 23

Camino 2: 1, 2, 3, 4, 6, 15, 16, 17, 22, 23

Camino 3: 1, 2, 3, 4, 7, 15, 16, 17, 22, 23

Camino 4: 1, 2, 3, 4, 8, 15, 16, 17, 22, 23

Camino 5: 1, 2, 3, 4, 9, 15, 16, 17, 22, 23

Camino 6: 1, 2, 3, 4, 10, 15, 16, 17, 22, 23

Camino 7: 1, 2, 3, 4, 11, 15, 16, 17, 22, 23

Camino 8: 1, 2, 3, 4, 12, 15, 16, 17, 22, 23

Camino 9: 1, 2, 3, 4, 13, 15, 16, 17, 22, 23

Camino 10: 1, 2, 3, 4, 14, 15, 16, 17, 22, 23

Camino 11: 1, 2, 3, 18, 19, 20, 21, 22, 23

3.5.1.2. Prueba de Caja negra.

La prueba de caja negra está en probar cada una de las funciones del sistema. Con esta prueba se busca las funciones que sean operativas.

Tabla 24

Prueba de Caja negra

ENTRADA	
Aprobación	Registro de Usuario. Registro de Eventos de Riesgo Operativo. Registro de Cuentas Contables. Registro de Tipo Evento. Registro de PAF. Registro de Canales. Registro de Proceso. Registro de Operación. Registro de Lugar. Registro de Líneas de Negocio. Registro de Incidentes de Seguridad.
Seguridad	Una vez teniendo usuarios, roles y contraseñas se podrá acceder.

FUNCIONES	
Software	Menú del jefe de sistemas (Administrador). Menú para el Encargado de Riesgos (Supervisor). Menú para el Oficial de Seguridad de la Información (Seguridad). Menú para el Técnico de Soporte e Infraestructura de Sistemas (Soporte).
SALIDA	
Resultado	Reporte específicos e individuales de cada registro. Reporte de Eventos de Riesgo Operativo individuales y por fechas requeridas. Reporte de Cuentas Contables individuales y por fechas requeridas. Reporte de Tipo Evento individuales y por fechas requeridas. Reporte de PAF individuales y por fechas requeridas. Reporte de Canales individuales y por fechas requeridas. Reporte de Proceso individuales y por fechas requeridas. Reporte de Operación individuales y por fechas requeridas. Reporte de Lugar individuales y por fechas requeridas. Reporte de Líneas de Negocio individuales y por fechas requeridas. Reporte de Incidentes de Seguridad individuales. Reporte de usuarios activos. Datos estadísticos del estado de registro y seguimiento de los Eventos de Riesgo Operativo. Datos estadísticos de la Cantidad de Eventos de Riesgo e Incidentes de Seguridad registrados.
Seguridad	Son resultados que verán los usuarios al acceder con su usuario y contraseña.

Nota. Prueba de Caja negra (Elaboración propia).

CAPÍTULO IV

CAPÍTULO IV
CALIDAD Y SEGURIDAD

4.1. INTRODUCCIÓN

En este capítulo se determina la calidad que posee el sistema, los procedimientos que se tiene que tomar en cuenta para temas de seguridad y la gestión de riesgos. La calidad del sistema es uno de los aspectos más importantes dentro del desarrollo de software del mismo modo se describirá la seguridad del sistema tomado en cuenta varios aspectos importantes. Debido a que el sistema ha sido desarrollado con una visión de calidad en el diseño y funcionalidad.

4.2. PRUEBAS DE CALIDAD

El desarrollo de la medición de calidad del software se realizará mediante la métrica ISO/IEC 9126 lo cual es un estándar internacional para la evaluación de software, que establece que cualquier componente de la calidad pueda ser descrito por las características de Funcionalidad, Confiabilidad, Mantenibilidad, Usabilidad y Portabilidad.

4.2.1. Funcionalidad

La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario.

Haremos uso de cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada. Los valores de los dominios de información se definen de la siguiente manera:

- ✓ **Números de entrada de usuario.** Se cuenta cada entrada del usuario que proporcione al Software.
- ✓ **Números de salida de usuario.** Se refiere a informes, pantallas, mensajes de error, etc.
- ✓ **Número de petición de usuario.** Se define como una entrada interactiva que resulta de algún tipo de respuesta en forma de salida iterativa.

- ✓ **Número de archivo.** Cuenta cada archivo maestro lógico, es decir que puede ser una parte de una base de datos o archivo independiente.
- ✓ **Número de interfaz externa.** Son las interfaces legibles por la máquina que utilizan para trasladar información a otro sistema.

Tabla 25

Cálculo de Punto de Fusión

Parámetros de Medición	Factores de ponderación		
	Cuenta	**Medio**	**Total**
N° de Entradas de Usuario	22	4	88
N° de Salidas de Usuario	17	5	85
N° de Peticiones de Usuario	26	4	104
N° de Archivos en Operación	7	10	70
N° de Interfaces Externos	0	7	0
Cuentas Total			**347**

Nota. Cuenta Total con Factor de Ponderación Medio (Elaboración propia).

Para calcular el punto de fusión se utilizará la siguiente ecuación.

$$PF = Cuenta\ Total * (Grado\ de\ Confiabilidad + tasa\ de\ error * \sum f_i)$$

Donde:

PF = Medida de Funcionalidad.

Cuenta Total = Es la suma de valor de las entradas, salida, peticiones, interfaces externas y archivos.

Grado de Confiabilidad = Es la confiabilidad estimada del sistema.

Tasa de error = Probabilidad subjetiva estimada del dominio de la información.

Σfi = Son valores de ajuste de complejidad según las respuestas destacadas en la siguiente tabla.

Tabla 26

Ajuste del factor de complejidad

N°	Factores de Complejidad	Valor
1	¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
2	¿Se requiere comunicaciones de datos?	4
3	¿Existen funciones procesamientos distribuidos?	3
4	¿Es crítico el rendimiento?	3
5	¿Sera ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?	4
6	¿Requiere el sistema entrada de datos interactiva?	4
7	¿Facilidad operativa?	4
8	¿Se actualizan los archivos maestros de forma interactiva?	3
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?	3
10	¿Es complejo el procedimiento interno?	4
11	¿Se ha diseñado el código para ser reutilizable?	4
12	Facilidad de instalación	4

13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	3
14	Facilidad de cambio	3
Total		**51**

Nota. Ajuste del factor de complejidad (Elaboración propia).

Reemplazamos la ecuación.

$$PF=347 * (0.65 + 0.01 * 51)$$

$$PF=402.52$$

Entonces:

$$PFMaxima=Cuenta\ Total*[0.65+0.01*\sum Fi]$$

El punto de fusión máximo que se puede alcanzar es:

$$PF=347 * (0.65+0.01*70)$$

$$PF=468.45$$

Resolviendo la curva normal

$$\text{Funcionalidad} = \frac{\text{PF}}{\text{PFMaxima}} * 100$$

Por lo tanto:

$$\text{Funcionalidad} = \left(\frac{402.52}{468.45}\right) * 100$$

$$\text{Funcionalidad} = 85.92\ \%$$

Por lo tanto, la funcionalidad del sistema es de 85.92%

4.2.2. Confiabilidad

La confiabilidad del sistema tiene la probabilidad de operación libre de fallos de un programa de computadora. La confiabilidad del software debe mantener su nivel de rendimiento bajo las condiciones establecidas por un periodo de tiempo, se mide de la siguiente manera.

La función siguiente muestra el nivel de confiabilidad del sistema:

$$\boldsymbol{F(t) = (Funcionalidad) * e^{-\lambda t}}$$

Se observa el trabajo hasta que se observa un fallo en un instante t, la función es la siguiente.

Probabilidad de hallar una falla: $P(T \leq t) = F(t)$

Probabilidad de hallar una falla: $P(T > t) = 1 - F(t)$

Valor de Funcionalidad previo = 85.92%

$\lambda = 0.01$ (es decir 1 error en cada 6 ejecuciones)

$t = 6\ meses$

Hallamos la confiabilidad del sistema

$$\mathrm{F}(12) = 85.92 * e^{-\frac{1}{6} * 12}$$

$$F(12) = 11{,}62\%$$

La probabilidad de hallar una falla es de un 11,62% durante los próximos 12 meses.

Por lo tanto, la probabilidad de no hallar un error es del 88.38%

4.2.3. Usabilidad

La usabilidad consiste en la evaluación del esfuerzo necesario que el usuario invertirá para usar el sistema, en base a su comprensión y estructura lógica que el sistema tiene. Usabilidad es la facilidad de uso, para determinar la usabilidad del sistema se usará la siguiente ecuación.

$$\mathrm{FU} = \frac{\left[\frac{\sum \chi_i}{n}\right] * 100}{\mathrm{N}}$$

Donde:

Xi = Sumatoria de valores

n = Numero de preguntas

N = Cantidad de personas

Tabla 27

Escala de valores "Usabilidad"

Escala	Valor
Muy buena	5
Buena	4
Media	3
Malo	2
Muy bajo	1

Nota. Escala de valores "Usabilidad" (Elaboración propia).

Tabla 28

Evaluación de preguntas para determinar la Usabilidad

N°	Pregunta	Evaluación
1	¿El sistema es fácil de utilizar?	5
2	¿El sistema facilita el trabajo que usted realiza?	4
3	¿Cómo considera los formularios que elabora el sistema?	4
4	¿El sistema tiene la seguridad necesaria?	4

5	¿Cómo considera el ingreso de datos del sistema?	5
	Total	22

Nota. Evaluación de preguntas para calcular la "Usabilidad" (Elaboración propia).

Se realizó la encuesta a 5 personas, así calculamos la facilidad de uso.

$$\mathrm{FU} = \frac{\left[\frac{22}{5}\right] * 100}{5}$$

$$\mathrm{FU} = 88$$

Por lo tanto, la usabilidad de uso es del 88%.

4.2.4. Mantenibilidad

Para hallar la mantenibilidad del sistema se utiliza el índice de madurez de software, que proporciona una indicación de la estabilidad de un producto de software (basado en los cambios que ocurren con cada versión del producto). Se calcula de la siguiente manera:

$$\mathrm{IMS} = \frac{[\mathrm{M}_t - (F_a + F_c + F_d)]}{M_t}$$

Donde:

Mt = Numero de módulos en la versión actual.

Fc = Numero de módulos en la revisión actual que se ha cambiado.

Fa = Numero de módulos en la versión actual que se han añadido.

Fd = Numero de módulos en la versión actual que se han eliminado.

Donde se obtendrá los siguientes valores:

Tabla 29

Valores obtenidos para determinar la Mantenibilidad

Información	Valores obtenidos
Mt	10
Fc	2
Fa	0
Fd	0

Nota. Valores obtenidos para determinar la "Mantenibilidad" (Elaboración propia).

Calculamos el índice de madurez del software los cuales son los resultados obtenidos en el sistema.

$$IMS = \frac{[10 - (0 + 2 + 0)]}{10}$$

$$IMS = 0.8 * 100\%$$

$$IMS = 80\%$$

Entonces el índice de madurez alcanzado es del 80%, este se encuentra en un rango satisfactorio.

4.2.5. Portabilidad

La portabilidad es la capacidad del sistema para ser trasladado de un entorno a otro. Para medir la portabilidad del software se usará la siguiente formula que indique el grado de portabilidad que tiene.

$$GP = 1 - \left(\frac{Número\ de\ días\ para\ portar\ el\ sistema}{Número\ de\ dias\ para\ implementar\ el\ sistema}\right)$$

Consideraciones:

1-. Para llevar el software a otro entorno solo se requiere copiar el código del sistema en una memoria extraíble con una capacidad de un 1GB.

Reemplazando los datos, se tiene:

$$GP = 1 - \left(\frac{1}{7}\right)$$

$$GP = 0.85 * 100$$

$$GP = 85\%$$

Por lo tanto, la portabilidad es de un 85%, que significa que el sistema es fácil de transportar, es decir puede ser llevado de un lugar a otro sin mucho esfuerzo.

4.2.6. Resultados

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente.

Tabla 30

Resultados de la Evaluación de Calidad - Norma ISO 9126

Características	Resultado (%)
Funcionalidad	85,92%
Confiabilidad	88,38%
Usabilidad	88%
Mantenibilidad	80%
Portabilidad	85%
Evaluación Total de Calidad	85.46%

Nota. Resultados de la Evaluación de Calidad - Norma ISO 9126 (Elaboración propia).

4.3. PRUEBAS DE SEGURIDAD

La seguridad de la información se puede definir como un conjunto de medidas, técnicas organizacionales y legales que puedan permitir a la organización proteger y asegurar su confidencialidad, integridad de la información que genera. La ISO – 27001 evalúa y rectifica la implementación mediante el cumplimiento de normas, así como la mejora continua de un conjunto de controles que permitan reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento de la institución en cuanto a la seguridad de la información, para lo cual se tomó los siguientes tipos de seguridad.

Seguridad a nivel de Base de Datos

En este proyecto de grado se hace uso del gestor de base de datos MySQL que proporciona estabilidad, confiabilidad y alto, además que brinda extensiones para distintas funcionalidades como ser encriptación de datos.

Trazabilidad

Procedimiento que permite registrar e identificar la ubicación y trayectoria del producto. En la trazabilidad de los registros en la base de datos, solo se muestran los atributos relevantes para la lógica del negocio.

Seguridad a nivel de aplicación

Tomando en cuenta las recomendaciones más relevantes especificadas en la norma ISO 27001 con respecto a la presentación de las características de confiabilidad, integridad y disponibilidad de la información se incorpora las siguientes medidas de seguridad del sistema.

Copias de Seguridad

Las copias de seguridad o backups son una medida esencial para la preservación de los datos informáticos.

Tabla 31

Medidas de Seguridad

Recomendaciones ISO 27001	Medidas de Seguridad incorporadas en el sistema
Control de Accesos	Se implementó como elemento importante la autenticación de usuario que consta de usuario y contraseña, el usuario deberá estar previamente autenticado para realizar cualquier acción, caso contrario será restringido al acceso de información.
Control criptográfico	Se implementó la encriptación de la contraseña de los usuarios con el uso de algoritmo de cifrado Hash.
Registro de actividad y supervisión	Se controla los registros de información mediante la validación de datos.

Nota. Medidas de Seguridad (Elaboración propia).

4.3.1. Autentificación y autorización

La autenticación y autorización van unidas por los accesos de usuario a distintos niveles de información. Este proceso realiza la autenticación de usuario tanto como los encargados o el administrador del sistema.

Figura 46

Autenticación y autorización

```
<div class="row justify-content-center">
    <div class="col-md-6 col-lg-4">
        <div class="login-wrap p-0">
            <div class="text-center">
                <h3 class="mb-4 text-center" style="color:#FFAD60; display: inline-block;font-weight: 500;">BIE
            </div>
        <form method="POST" action="{{ route('login') }}" class="signin-form">
                @csrf

            <div class="form-group">
                <input type="text" name="name" class="form-control" placeholder="Usuario" required>
            </div>
        <div class="form-group">
          <input id="password-field" type="password" name="password" class="form-control" placeholder="Contraseña"
          <span toggle="#password-field" class="fa fa-fw fa-eye field-icon toggle-password"></span>
        </div>
        <div class="form-group">
            <button type="submit" class="form-control btn btn-primary submit px-3">Entrar</button>
        </div>
```

Nota. Autenticación y autorización (Elaboración propia).

4.3.2. Encriptación

Para la encriptación se utilizará el algoritmo HASH, que nos permite encriptar la contraseña y la verificación se realiza comparando encriptaciones.

Figura 47

Encriptación

```
    public function create(array $input)
    {
        Validator::make($input, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => $this->passwordRules(),
            'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ? ['accepted', 'required'] : '',
        ])->validate();

        return User::create([
            'name' => $input['name'],
            'email' => $input['email'],
            'password' => Hash::make($input['password']),
        ]);
    }
```

Nota. Encriptación (Elaboración propia).

4.3.3. Copias de Seguridad - Backups

Gracias a las copias de seguridad, conseguiremos tener un plan de acción en caso de que se produzca un problema con los sistemas de la Cooperativa, en caso de perder parte o toda la información, se podrá recuperar rápidamente.

Figura 48

Copias de Seguridad - Backups

```
<div class="container-fluid">
    <div class="d-sm-flex align-items-center justify-content-between mb-4">
        <h1 class="h3 mb-0 text-gray-800">{{ 'Copiad de seguridad' }}</h1>
        <a class="btn btn-sm btn-primary" href="{{ route('backup.create') }}">Crear Copia de Seguridad</a>
    </div>
    <div>

    @if(session()->has('info'))
    <div>
        <h3>{{ session('info') }}</h3>
    </div>
    @endif
    </div>
 <div class="card shadow mb-4">
<div class="card-body">
    <div class="table-responsive">
        <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
            <thead>
                <tr>
                    <th>Nombre</th>
                    <th>Fecha de creacion</th>
                    <th>Tamaño</th>
                    <th>Acciones</th>
                </tr>
            </thead>
            <tbody>
                @foreach($db_zips as $db_zip)
                <tr>
                    <td>{{ $db_zip['name'] }}</td>
                    <td>{{ $db_zip['time'] }}</td>
                    <td>{{ $db_zip['tamaño'] }}</td>
                    <td>
                        <form action="{{ route('backup.downloadZip') }}" method="POST" class="d-inline">
                            @csrf
                            <input type="text" name="filenamee" class="collapse" value="{{ $db_zip['name'] }}">
                            <button class="btn btn-sm btn-primary" type="submit">Descargar</button>
                        </form>
```

Nota. Copias de Seguridad - Backups (Elaboración propia).

CAPÍTULO V

CAPÍTULO V
COSTO BENEFICIO

5.1. INTRODUCCIÓN

En el presente capitulo se aplicará los parámetros del modelo COCOMO para determinar el valor económico del software. Estos métodos no son otra cosa que establecer una relación matemática entre el esfuerzo y el tiempo de desarrollo. Para lo cual se utiliza el método algorítmico de aproximación COCOMO II, orientado a los puntos de función ajustados.

5.1.1. Método de estimación de costos COCOMO II

En el ámbito de la ingeniería de software, la estimación de costos radica básicamente en estimar la cantidad necesaria de personas para hacer el desarrollo de software. A diferencia de otras disciplinas de la ingeniería, en las cuales, el costo de los materiales es el primer componente a ser estimado, así que para el proyecto se utilizará COCOMO II.

Donde el método es el más utilizado, y donde trabaja en función del tamaño del software tomando como parámetro las líneas de código, interpretadas en KLDC es decir kilo líneas de código. Al analizar las líneas de código de mi Proyecto de Grado denominado “Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad” Caso: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. y realizar la sumatoria de todos los lenguajes, framework, base de datos.

Líneas de código del proyecto: 6106

$$KLDC = \frac{LDC}{1000}$$

$$KLDC = \frac{6106}{1000}$$

$$KLDC = 6.106$$

Con el primer dato obtenido KLDC se empezará a estimar el costo y los beneficios, tras lo cual se utilizará las siguientes variables, ecuaciones, en unidades y tiempo.

Tabla 32

Ecuaciones para calcular el modelo COCOMO II

VARIABLE	ECUACION	TIPO/UNIDAD
Esfuerzo requerido	$E = a(KLDC)^b * FAE$	Personas /mes
Tiempo Requerido	$T = c * (E)^d$	Meses
Número de personas	$NP = \frac{E}{T}$	Personas
Costo Total	$\boldsymbol{CT = Sueldo\ Mes * NP * T}$	**$us Bs.**

Nota. Ecuaciones para calcular el modelo COCOMO II (Elaboración propia)

Donde:

E = es el esfuerzo requerido por el proyecto, en personas por mes.

T = es el tiempo requerido por el proyecto, en meses.

NP = es el número de personas que requiere el proyecto.

a, b, c, y d son las constantes con valores definidos, según cada sub modelo.

KLCD = es la cantidad de líneas de código, expresados en miles.

FAE = es el multiplicador que depende de la tabla de los 15 atributos.

A la vez cada modelo se subdivide en modos, los mismos son:

- ✓ **Modo orgánico:** Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- ✓ **Modo semi – libre o semi acoplado:** Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- ✓ **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a

resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

Tabla 33

Coeficientes del modelo COCOMO II

MODO	A	B	C	D
Orgánico	2.40	1.05	2.50	0.38
Semiacoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Nota. Coeficientes del modelo COCOMO II (Elaboración propia)

Para el caculo de variable FAE utilizaremos la tabla de 15 atributos, para el desarrollo del análisis de costo.

Tabla 34

Cálculo de atributos FAE

Atributos	Valor					
	Muy bajo	**Bajo**	**Nominal**	**Alto**	**Muy alto**	**Extra alto**
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	-
Tamaño de Base de datos	-	0,94	1,00	1,08	1,16	-
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución	-	-	1,00	1,11	1,30	1,66
Restricciones de memoria virtual	-	-	1,00	1,06	1,21	1,56

Volatilidad de la máquina virtual	-	0,87	1,00	1,15	1,30	-
Tiempo de respuesta	-	0,87	1,00	1,07	1,15	-
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	-
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	-
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	-
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90	-	-
Experiencia en el lenguaje	1,14	1,07	1,00	0,95	-	-
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	-
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	-
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	-
Total			0,87			

Nota. Cálculo de atributos FAE (Elaboración propia)

Calculando el esfuerzo:

$$E = a(KLDC)^b * FAE$$

$$E = 2.40\ (6.106)^{1.05} * 0.87$$

$$E = 13.95$$

- $E = 13.95$ (personas/mes)

Calculando el tiempo de desarrollo:

$$T = c * \ (E)^d$$

$$T = 2.50 * \ (13.95)^{0.38}$$

$$T = 6.81$$

- $T = 6.81$ (meses)

Calculando promedio:

$$NP = \frac{E}{T}$$

$$NP = \frac{13.95}{6.81}$$

$$NP = 2.05$$

- $NP = 2.05$ (personas)

Estimando el salario promedio mínimo actual de un profesional programador Bs. 3300, número que será tomado en cuenta para la estimación:

$$CT = Sueldo\ Mes * NP * T$$

$$CT = 3300 * 2 * 7$$

$$CT = 46200$$

- $CT = 46200$ Bs.

En resumen, se requiere 2 personas estimado un trabajo de 7 meses para el desarrollo del sistema con un costo total de Bs. 46200.

CAPÍTULO VI

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Después de haber finalizado con el desarrollo del proyecto tomando en cuenta la problemática inicial y los objetivos planteados se puede afirmar que se ha alcanzado la meta trazada y pruebas necesarias del “Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad” Caso: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. logrando los requerimientos de la institución.

El entorno tecnológico para desarrollar el sistema, resulto muy apropiado y no presentaron inconvenientes importantes durante el desarrollo del mismo, que podrían contradecir la elección realizada.

Tabla 35

Objetivos y Conclusiones

OBJETIVOS	CONCLUSIONES
Objetivo general. Desarrollar un Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad para cumplir con lo que establece la ASFI., en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.	Se desarrolló un sistema para el seguimiento y control del Riesgo Operativo e Incidentes de Seguridad, mismo que cumple con lo que establece la ASFI., en la Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.
○ No tiene desarrollado e implementado un Sistema de Información para la gestión de Riesgo Operativo.	○ Se desarrollo un sistema de información para la Gestión de Riesgo Operativo.

○ No cuenta con los reportes correspondientes a la Central de Información de Riesgo Operativo (CIRO).	○ Con la Implementación del software se obtuvieron los reportes correspondientes a la Central de Información de Riesgo Operativo.
○ La entidad no efectuó un diagnóstico de seguridad física que identifique los niveles de riesgo ante incidentes de seguridad.	○ Se desarrollo un módulo para los Incidentes de Seguridad del cual se obtienen los reportes para identificar los niveles de riesgo.
• Realizar la recopilación de información para conocer el flujo de los datos dentro de la Cooperativa y obtener puntualidad a la hora de su entrega.	Se realizó la recopilación de información y se armó un diagrama de flujo en relación al Riesgo Operativo e Incidentes de Seguridad.
• Centralizar la información de los eventos de riesgo operativo e incidentes de seguridad de la información en una Base de Datos para optimizar tiempos y dar una solución oportuna.	Se centralizó la información de los eventos de riesgo operativo e incidentes de seguridad de la información en una Base de Datos.
• Automatizar perfiles de usuario para ordenar la forma de atención que se dará a los eventos o incidentes registrados.	Se automatizó perfiles de usuario para ordenar la forma de atención que se da a los eventos o incidentes registrados.

• Sistematizar el seguimiento de datos para controlar el estado del evento o incidente hasta darle solución	Se sistematizó el seguimiento de datos para controlar el estado del Riesgo Operativo e Incidentes de Seguridad.

Nota. Objetivos y Conclusiones (Elaboración propia)

6.2. RECOMENDACIONES

Al igual que el avance de la tecnología se evidencia la evolución de los sistemas. En base a las observaciones realizadas en el periodo de desarrollo dentro de la Cooperativa USAMA, las recomendaciones que se deben de considerar en el "Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad" son los siguientes:

- ✓ Capacitar a todos los nuevos usuarios, para que puedan realizar operaciones en el sistema y así poder administrarlos correctamente.
- ✓ Se recomienda cambiar continuamente las contraseñas para seguridad del sistema.
- ✓ El área de Sistemas debe realizar copias de seguridad en periodos semanales o mensuales para resguardar toda la información en caso de un problema técnico externo.

Bibliografía

Abud Figueroa, M. A. (2012). *Calidad en la Industria del Software. La Norma ISO-9126*. 1–2.

Akly Aguayo, F. (2019). *Modelo de Seguridad basado en la Norma ISO/IEC 27001/2013, para minimizar los Riesgos en la Seguridad Logica de la Informacion*. 18.

Alejandro Montoya, M. (2000). *Copias de seguridad*. 4.

Ambros Mendioroz, M. (2017). *Sistema de Gestion de Incidencias*. 12–13.

Anibarro Zelaya, C. E. (2001). *Manual Básico de HTML*. 6.

Anonimus. (1955). *Ciclo de vida de sistemas y sus etapas*. 5.

Aramayo F., M. M. (2006). *Gestion de Riesgos Cooperativa de Ahorro y Credito Abierta "San Jose de Bermejo" Ltda*. 21–30. https://doi.org/10.1002/ejoc.201200111

Aramayo Shaw, I. A. (2017). *Guia Metodologica de Gestion de Riesgo Operacional para Instituciones Financieras de Desarrollo de Bolivia*. 22.

Area de Tecnologias de la Informacion y las Comunicaciones Aplicadas. (2010). *Manual Basico de creacion de Paginas Web*. 9.

Arias Chaves, M. (2006). *La ingenieria de requerimientos y su importancia en el desarrollo de proyectos de software*. 3–4.

Asociacion de Academias de la Lengua Española. (2021). Operativo. In *Real Academia Española*. https://dle.rae.es/operativo

Bertalanffy, V. (2010). *Teoria general de sistemas*. 7.

Bonifacio Carrasco, K. V., & Velasquez Rojas, J. A. (2020). *Riesgo Operativo y su Influencia en las decisiones Financieras en las cajas de Ahorro y Credito en Lima provincia, Año 2018*. 15.

Callisaya Corina, J. C. (2015). *Sistema Integral Web de Promocion Institucional para desarrollo social y economico*. 68-69–70.

Canepa Sáenz, A. A., & García González, C. E. (2011). *Comparativas de los Modelos de Ciclo de Vida*. 7.

Cantone, D. (2006). *La Biblia Del Programador Implementacion Y Debugging*. 16.

Casillas Santillan, L. A., Gibert Ginesta, M., & Perez Mora, O. (2014). *Bases de datos en MySQL*. 5.

Castilla-La Mancha. (2021). *¿Qué es una Cooperativa?* 1. https://www.produce.gob.pe/index.php/cooperativas/que-es-una-cooperativa

Chavez Gonzales, D. (2012). *Manual de usuario del Open Journal Systems*. 1. https://wordpress.com/es/support/roles-usuario/

Chiavenato, I. (2007). *Introduccion a La Teoria General De La Administracion*.

CIVICUS. (2007). *Seguimiento y Evaluación Seguimiento y Evaluación*. 3.

Cobo, A., Gomez, P., Perez, D., & Rocha, R. (2005). *PHP y MySQL Tecnologías para el desarrollo de aplicaciones web*. 36.

Consejo para las Tecnologias de Informacion y Comunicacion. (2010). *Guía para la Gestión de Incidentes de Seguridad de la Información*. 3.

COPLUTIC y AGETIC. (2017). *Plan de implementación de gobierno electrónico 2017 – 2025*. 20.

Coronel Delgado, P. X., & Peralta Espinoza, P. A. (2009). *Diseño e implementacion de un sistema PID para el control de nivel de un tanque desarrollado con el PLC siemens S7-200*. 11.

Cortés, A. M. (2017). *Reporte e investigación de incidentes y accidentes de trabajo*. 3.

Cotaña, M. (2014). *Análisis y Diseño de Sistemas de Información*. 6-7-8–9.

Deheza Salinas, A. S. (2019). *Riesgo Operativo en el proceso de Emision de Giros al Exterior*. 13.

Echemendía Tocabens, B. (2011). *Definiciones acerca del riesgo y sus implicaciones*. 2.

Eguiluz Pérez, J. (2008). *Introduccion a CSS*. 5.

ESET Enjoy Safer Technology. (2014). *Guia corporativa Cifrado de la información*. 6.

Evidian. (2011). *Los 7 Metodos de Autentificación más utilizados*. 6.

Facultad de tecnologia de la Información. (2002). *Aplicacion del modelo ISO 9126 para la evaluacion de un sistema de aprendizaje virtual*. *2*, 4.

Fernández, L., Pérez, M., Menéndez, M., & Lázara, M. (2008). *Accidentes e incidentes de trabajo*. http://www.ccoo.cat/pdf_documents/aatt.pdf

Fernandez Mamani, B. (2016). *Sistema de Seguimiento y Control de Actividades Mineras*. 41.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*.

González González, E., & Galarza Galarza, M. (2016). *Desarrollo de una pagina web Infantil en HTML5 y Boostrap*. 17.

Haverbeke, M. (2018). *Eloquent JavaScript*.

Hostalia Whitepapers. (2003). *Laravel, un framework de PHP*. 3.

Huerta de los Santos, A., & Muñoz Serafin, M. (2018). *Boostrap*. 17.

INEC. (2017). *Memorias del Sismo: Reconstruyendo las cifras*. 16. https://www.ecuadorencifras.gob.ec/memorias-del-sismo-reconstruyendo-las-cifras/

INEI Instituto Nacional de Estadistica e Informatica. (2010). *Las cooperativas en el Perú*. 14. https://www.aciamericas.coop/IMG/pdf/cooperu.pdf

Laudon, K. C., & Laudon, J. P. (2004). *Sistemas de información gerencial*. 54.

López Rosciano, R. A., & Pech Montejo, J. A. (2015). Desarrollo de herramienta de gestión de proyecto RUP usando metodología SCRUM + XP. *Universidad Politecnica de Madrid*, 6-10–11. http://oa.upm.es/44208/3/TFM_RODRIGO_ANTONIO_LOPEZ_ROSCIANO_JOSE_ALFREDO_PECH_MONTEJO.pdf

Maida, E. G., & Pacienzia, J. (2015). *Metodologías de desarrollo de software*. 18.

María del C. Lopez, A. G., & Alejandra Otazú, S. M. (2013). *Un Modelo de Estimacion de Proyectos de Software*. 26.

Márquez, E. (2015). *Clasificación de los Sistemas*. 1.

MDS-ALFA. (2009). *Metodologia de Desarrollo de Sistemas ALFA*. 5.

Mega Practical. (2010). *Metodologias de Desarrollo de Software*. 4–5.

Mendoza Tellez, X. (2020). *Informe Seguimiento del Plan de Accion 2019* (pp. 11–12).

Montero Bagatella, J. C. (2013). *El concepto de seguridad en el nuevo paradigma de la normatividad mexicana*. 3.

Muncha Quezada, A. C. (2018). *Administración del riesgo operativo y los efectos económicos-financieros antes y posterior a la normativa de la gestión del riesgo operativo*. *documento*, 8.

Mutual de Seguridad. (2015). *Investigacion de Incidentes/ Accidentes*. 3. http://www.ingenieroambiental.com/4023/62_investigacion incidente accidente.pdf

Nuñez Mora, J. A., & Chavez Gudiño, J. J. (2010). *Riesgo operativo: esquema de gestión y modelado del riesgo*. 4.

Oz, E. (2008). *Administración de los sistemas de información*. https://www.emagister.com/uploads_user_home/Comunidad_Emagister_8601_laudon.pdf

Pelissier Q., C. (2002). *Programación con PHP*. 5.

Penalva Lucas, J. L. (2017). *El concepto de seguridad. Importancia relativa de las dimensiones de la seguridad en Europa y en el sur del Mediterráneo*. 3.

Pérez Garcia, A. A. (2007). *Desarrollo de herramientas web de gestión docente*. 13–14.

Pressman, R. S. (2005). Ingeniera del Software V Edicion. In *CITEG Revista Arbitrada* (Vol. 1, Issue 5).

Pressman, R. S. (2010). Ingeniería de Software un enfoque práctico VII Edicion. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 7).

Quispe Nina, J. W. (2017). *Gestion de Riesgo Operativo para el Banco FIE S.A. sobre la Base de Normativa Vigente de la ASFI*. 14.

Rodas Palomeque, P., & Ulloa Brito, L. (2006). *Estudio practico sobre la seguridad de los datos con el gestor de base de datos Microsoft Sql Server 2000*. 18.

Sanchez Asenjo, J. (2013). *CSS*. 7–8.

Sánchez, J. (2011). *Sistemas de Autenticación Y Autorización En Internet*. *1*, 4.

Sanchez Pedros, J. (2017). *Desarollo de un Sistema Web para la gestion de Imagenes y Albumes*. 12–13.

Sanchez Peño, J. M. (2015). *Pruebas de Software. Fundamentos y Técnicas.* 29-30-37–38.

Sans Securing The Human. (2011). *Boletin mensual de consejos de Seguridad para usuarios de Computadoras*. 1.

Sanz, E. (2016). *Consultores Documentales*. 1. https://sorprendemos.com/consultoresdocumentales/?p=507#:~:text=El seguimiento consiste básicamente en,lo planificado y esperado ocurra

Seguridad y Privacidad de la Informacion. (2009). *Guía para la Gestión y Clasificación de Incidentes de Seguridad de la Información.* 19.

Soldano, Á. (2009). *Conceptos sobre Riesgo*. 2. http://www.rimd.org/advf/documentos/4921a2bfbe57f2.37678682.pdf

Sommerville, I. (2005). *Ingeniería del software*.

Thompson, I. (2008). *Definición de Información*. 2.

Torossi, G. (2002). *El Proceso Unificado de Desarrollo de Software*. 3-5-7-8–9.

Uturunco Mamani, W. N. (2022). *Gestion de Riesgo Operativo en las Instituciones Microfinancieras de Bolivia en el periodo 2006-2018*. 20.

Valdez Alvarado, V. (2012). *Técnicas efectivas para la toma de requerimientos*. 4–5.

Van de Velde, H. (2009). *Sistemas de Evaluación, Monitoreo, Seguimiento y Evaluación de Proyectos Sociales*. http://abacoenred.com/wp-content/uploads/2016/01/Sistemas-de-Evaluación-Monitoreo-Seguimiento-Evaluación-III-edición.pdf.pdf

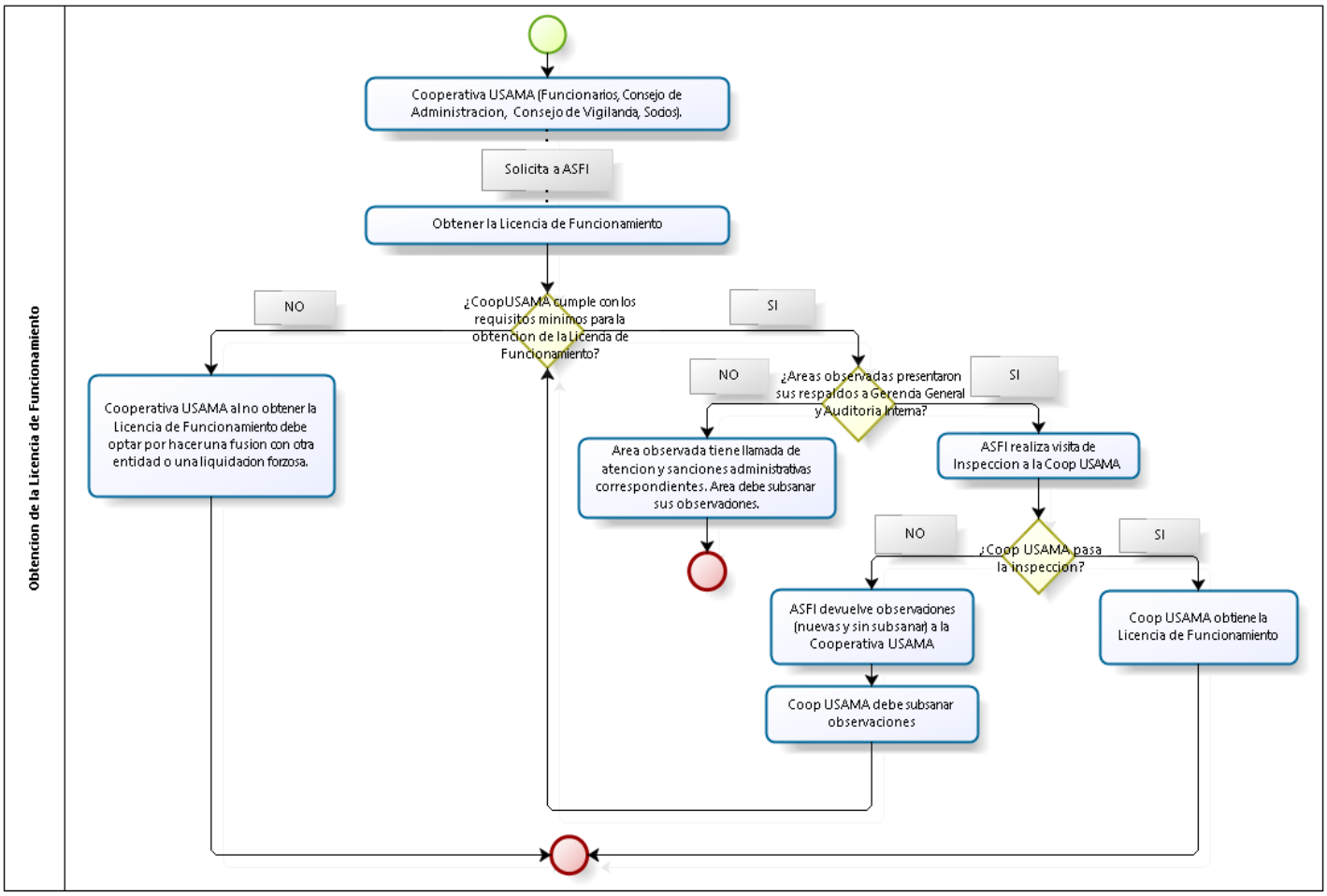
Velezmoro La Torre, O. A. (2010). *Modelo de Gestion de Riesgo Operacional en una Institucion Financiera Peruana dentro de un enfoque integrado de Gestion de Riesgos*. 37. http://tesis.pucp.edu.pe/repositorio/handle/123456789/4826

Yuri Cabrera, E. (2016). *Control*. 1–2.

Anexos

Figura 49

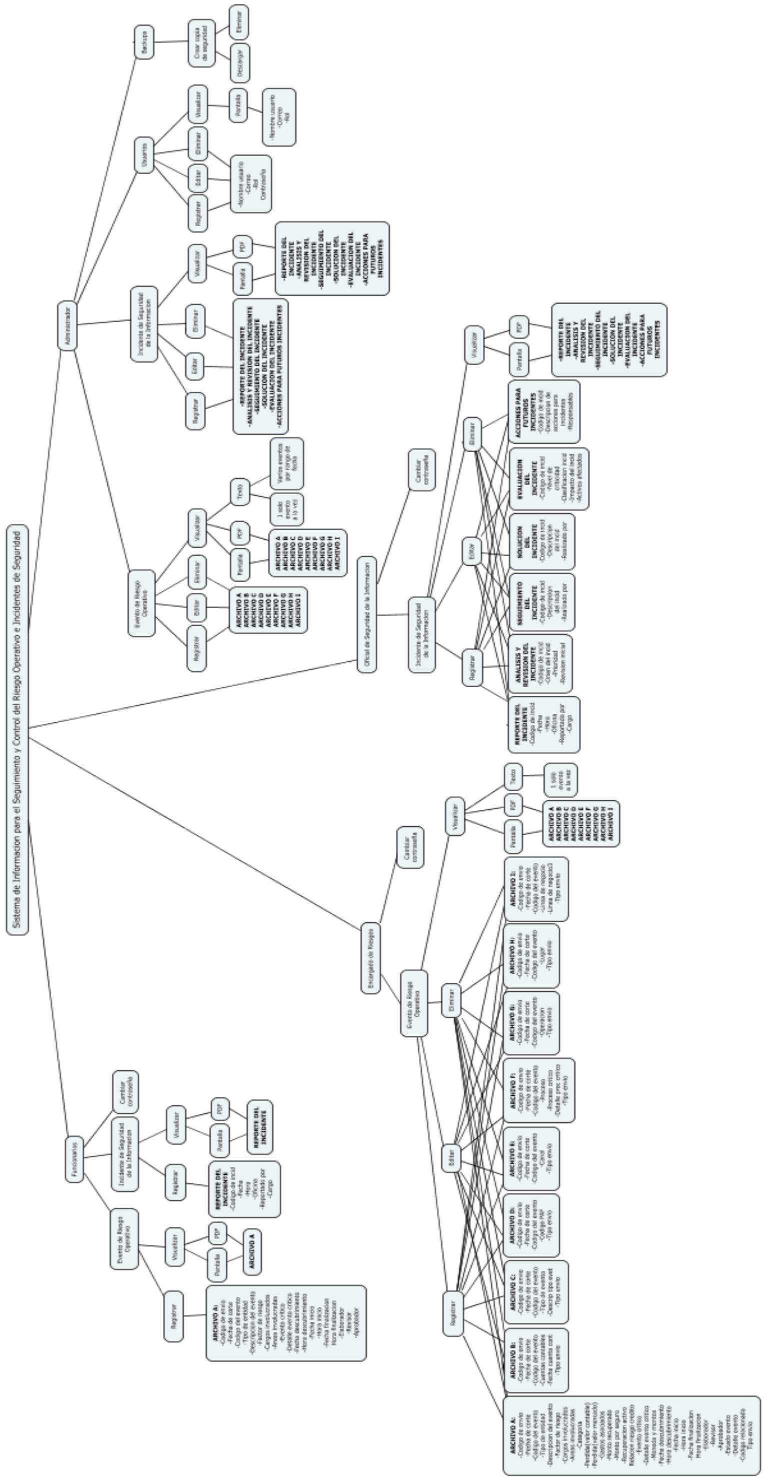
Diagrama de Flujo - Obtención de la Licencia de Funcionamiento



Nota. Diagrama de Flujo - Obtención de la Licencia de Funcionamiento dentro de la Cooperativa Unión Santiago de Machaca USAMA Ltda. (Actualmente).

Figura 50

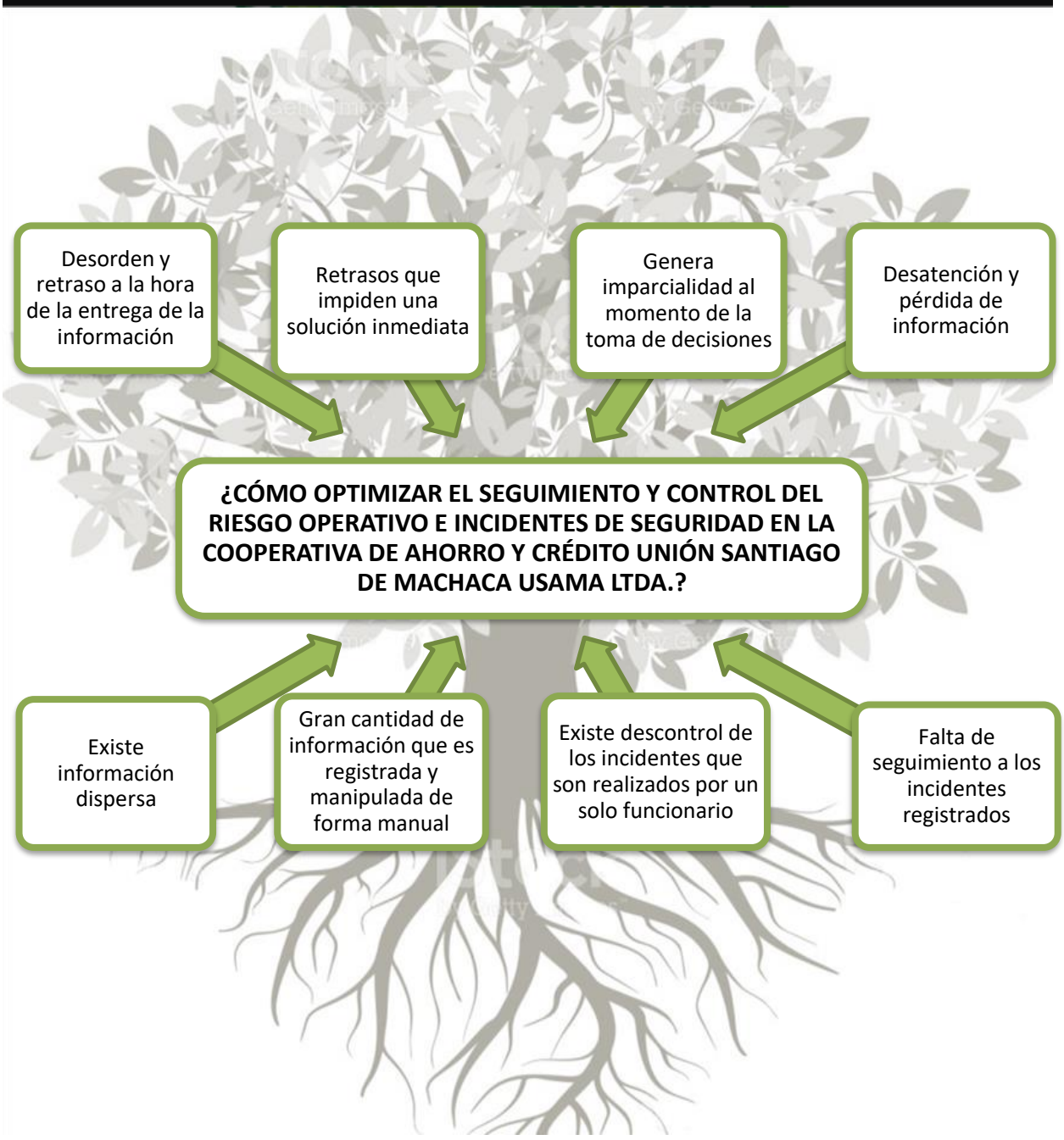
Propuesta de Sistema (Módulos)



Nota. Esquema – Propuesta de Sistema por módulos: funcionarios, Encargado de Riesgos, Oficial de Seguridad de la Información y Administrador para la Cooperativa Unión Santiago de Machaca USAMA Ltda.

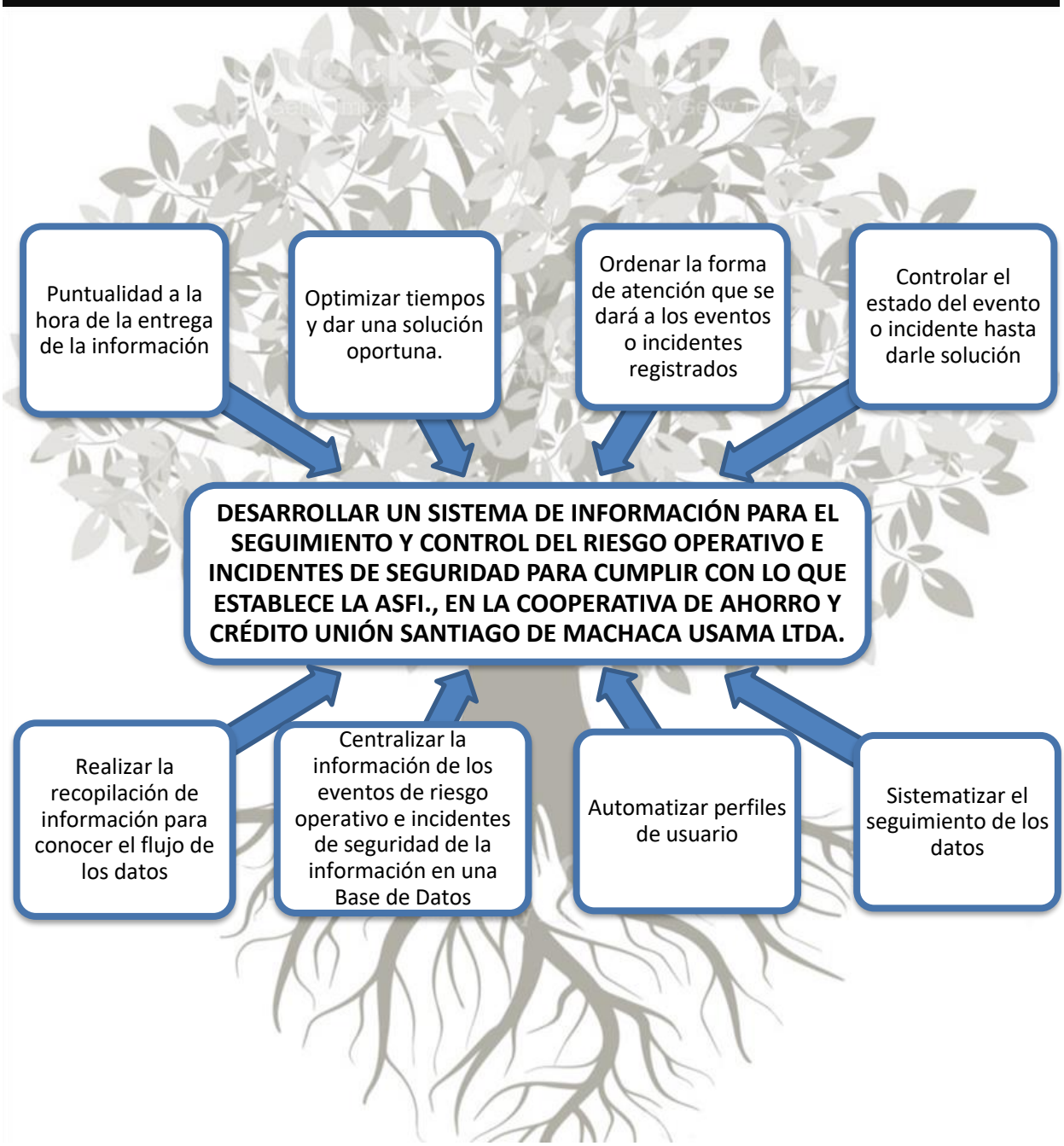
ANEXO A

Árbol de Problemas



ANEXO B

Árbol de Objetivos



ANEXO C

USAMA COOP
COOPERATIVA DE AHORRO Y CREDITO
Funfdada el 19 de Julio de 1985

MANUAL PARA EL ROL DE OPERADOR

INFOSYS

- CENTRAL DE INFORMACIÓN DE RIESGO OPERATIVO
- INCIDENTES DE SEGURIDAD DE LA INFORMACIÓN

INDICE

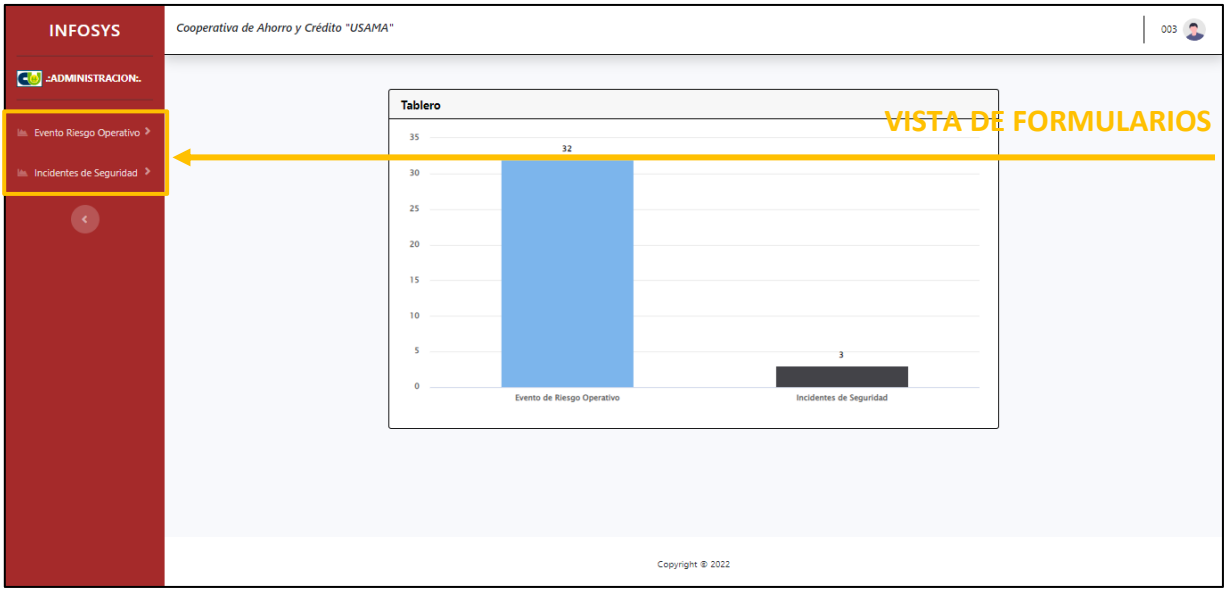
MANUAL PARA EL ROL DE OPERADOR INFOSYS

1. OBJETIVO

El presente manual tiene por objeto establecer el llenado correcto de la información de los Eventos de Riesgo Operativo y los Incidentes de Seguridad de la Información, por parte del rol Operador que registra el evento de manera inicial.

2. DESCRIPCIÓN DEL SISTEMA INFOSYS PARA ROL OPERADOR

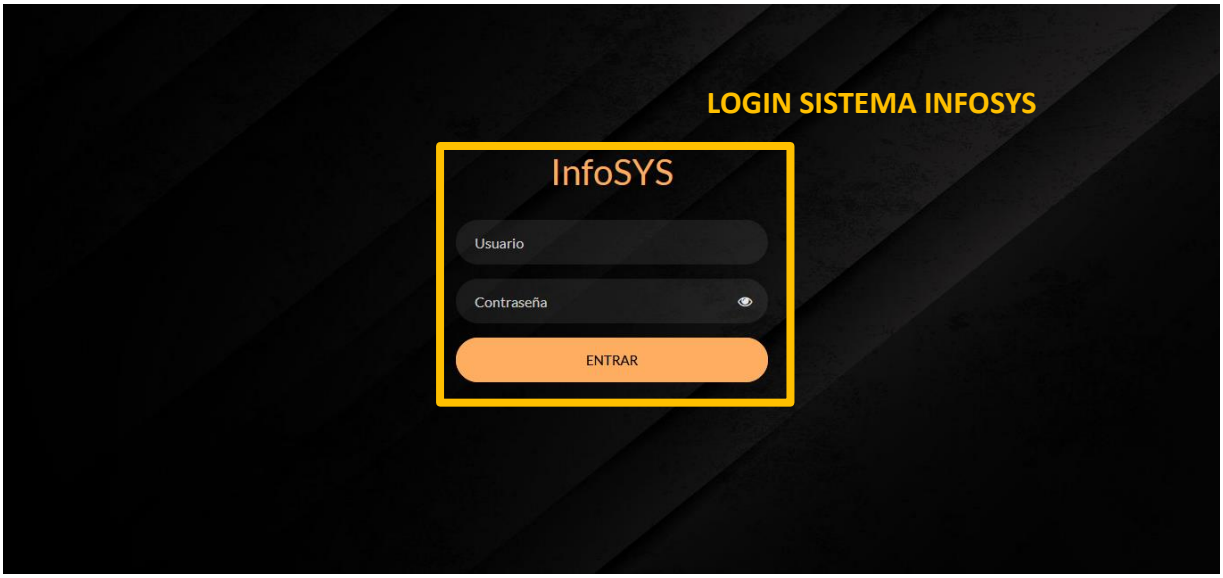
El sistema INFOSYS para el rol de Operador está conformado por 1 formulario para el registro de los Eventos de Riesgo Operativo y un segundo formulario para el registro de los Incidentes de Seguridad de la Información redistribuida estructuralmente.



INGRESO AL SISTEMA

Para que el rol de Operador pueda llenar los eventos de Riesgo Operativo se debe realizar los siguientes pasos:

- Paso 1 : Ingresar al link http://192.168.1.70/infosys/
- Paso 2 : El rol de Operador debe ingresar los siguientes datos:
  - Usuario : Usuario que se asignó vía correo electrónico
  - Contraseña : La contraseña del rol Operador asignado vía correo



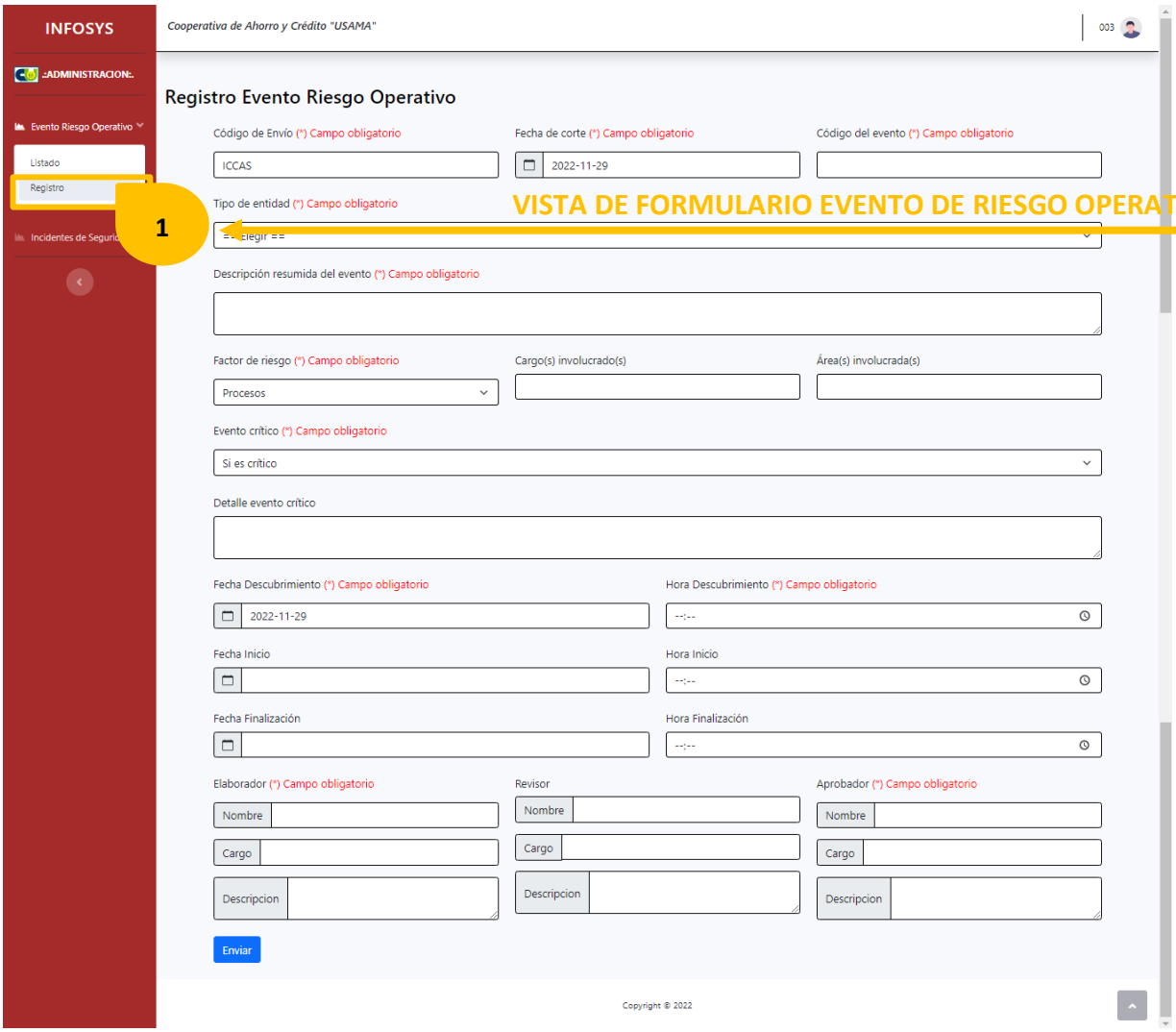
El formulario cuenta con campos redistribuidos estructuralmente y ordenados según para el envío de información electrónica a la Central de Información de Riesgo Operativo, los cuales facilitan interactuar al usuario con el Encargado de Riesgos y el Oficial de Seguridad de la Información.

3. FORMULARIO PARA REGISTRAR EVENTO DE RIESGO OPERATIVO

En este formulario se procede de la siguiente manera:

- Reportar desde el rol de Operador con datos generales del evento de riesgo operativo.
- Para ingresar a este formulario deberá ingresar a la OPCION: **"Registro"**.

Posterior a la selección de la OPCION, el sistema le direccionará al formulario correspondiente:



3.1. Código de Envío.

Código Identificador único asignado por la entidad ASFI, para el envío de información.

3.2. Fecha de Corte.

Fecha de corte a la que corresponde la información reportada.

3.3. Código de evento.

Identificador único que da la Entidad para cada evento de riesgo.

3.4. Tipo de Entidad.

Identifica el tipo de entidad reportante según Lista.

3.5. Descripción resumida del Evento.

Descripción del evento suscitado.

3.6. Factor de riesgo.

Corresponde a la fuente, causa primaria o el origen de un evento de riesgo operativo.

3.7. Cargos involucrados.

Descripción del cargo(s) de la(s) persona(s) afectadas o involucradas por el evento de riesgo operativo.

3.8. Áreas involucradas.

Descripción de la(s) área(s) afectadas o involucradas por el evento de pérdida.

3.9. Evento crítico.

Identifica si el evento se considera como crítico o no.

3.10. Detalle Evento crítico.

Contiene el detalle de la razón para considerar el evento como crítico.

3.11. Fecha de descubrimiento.

Campo de fecha, es obligatorio, indica la fecha en la cual ha sido descubierto el evento de riesgo. En el formato día/mes/año.

3.12. Hora de descubrimiento.

Campo de hora, es obligatorio, indica la hora en la cual ha sido descubierto el evento de riesgo. En el formato horas: minutos.

3.13. Fecha de inicio.

Campo de fecha, indica la fecha donde ha iniciado el evento, En el formato día/mes/año.

3.14. Hora de inicio.

Campo de hora, indica la hora donde ha iniciado el evento, en el formato horas: minutos.

3.15. Fecha de fin.

Campo de fecha, indica la fecha donde ha terminado el evento, En el formato día/mes/año.

3.16. Hora de fin.

Campo de hora, indica la hora donde ha terminado el evento, en el formato horas: minutos.

3.17. Elaborador.

Responsable de la elaboración del reporte a la CIRO. Se debe indicar el nombre, cargo y una breve descripción de las funciones que desempeña en la entidad.

3.18. Revisor

Responsable de la revisión del reporte a la CIRO. Se debe indicar el nombre, cargo y una breve descripción de las funciones que desempeña en la entidad.

3.19. Aprobador

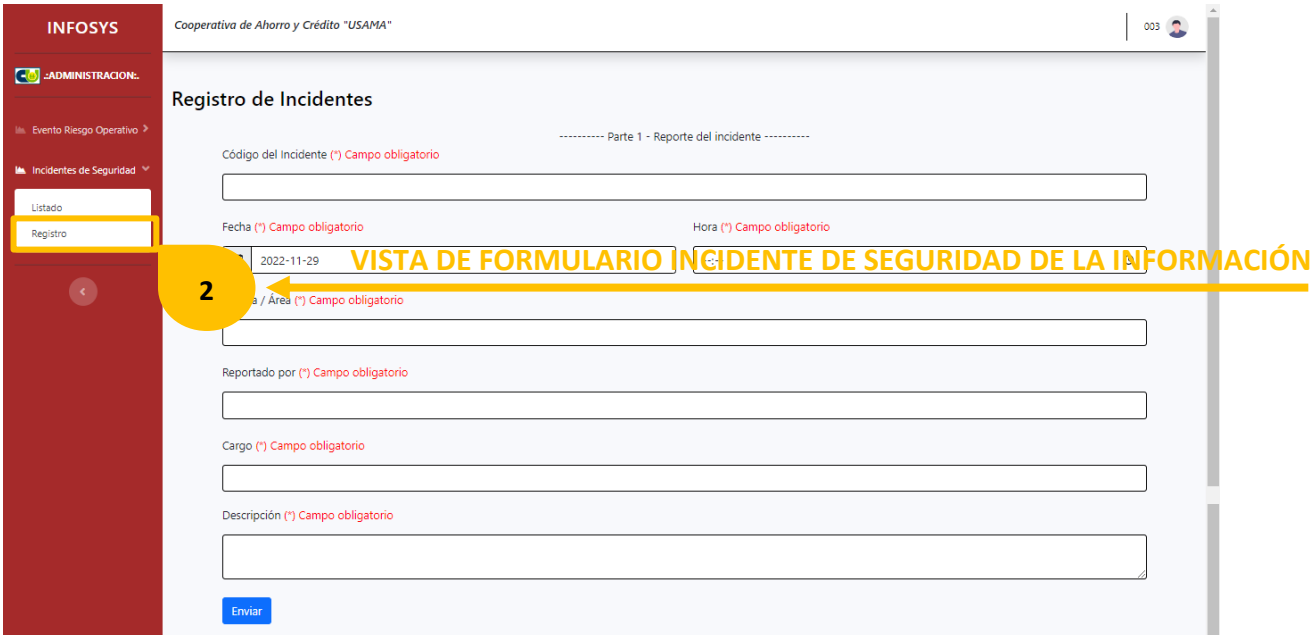
Responsable de la aprobación del reporte a la CIRO. Se debe indicar el nombre, cargo y una breve descripción de las funciones que desempeña en la entidad.

4. FORMULARIO PARA REGISTRAR INCIDENTE DE SEGURIDAD DE LA INFORMACIÓN

En este formulario se procede de la siguiente manera:

- Reportar desde el rol de Operador con datos generales del incidente de seguridad de la información.
- Para ingresar a este formulario deberá ingresar a la OPCION: **"Registro".**

Posterior a la selección de la OPCION, el sistema le direccionará al formulario correspondiente:



4.1. Código de Incidente.

Código Identificador único asignado por la entidad.

4.2. Fecha.

Fecha de corte a la que corresponde la información reportada.

4.3. Hora.

Hora de corte a la que corresponde la información reportada.

4.4. Oficina/Área.

Descripción de la Oficina afectadas o involucradas por el incidente de seguridad.

4.5. Reportado por.

Responsable de la elaboración del reporte.

4.6. Cargo.

Se debe indicar el cargo.

4.7. Descripción.

Descripción del incidente suscitado.

5. CAMBIAR CONTRASEÑA

En este menú podemos realizar el cambio de la contraseña de nuestro usuario.

ANEXO D

USAMA COOP

COOPERATIVA DE AHORRO Y CREDITO

Funfdada el 19 de Julio de 1985

MANUAL TECNICO

INFOSYS

- CENTRAL DE INFORMACIÓN DE RIESGO OPERATIVO
- INCIDENTES DE SEGURIDAD DE LA INFORMACIÓN

INDICE

MANUAL TÉCNICO INFOSYS

1. OBJETIVO

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo. Informar y especificar al usuario la estructura y conformación del sistema con el fin de que se pueda hacer soporte y modificaciones o actualizaciones al sistema en general.

2. OBJETIVOS ESPECÍFICOS

- Definir claramente el proceso de instalación del aplicativo.
- Detallar los requerimientos mínimos de Hardware y Software para el funcionamiento del aplicativo.
- Describir las herramientas utilizadas en el desarrollo del aplicativo.

3. INTRODUCCIÓN

En este manual se describe los pasos necesarios para poner en funcionamiento el aplicativo, se requiere que la persona encargada de la instalación, tenga conocimientos básicos de sistemas.

4. PROCESOS

4.1. Procesos de Entrada

- Rol Operador
    - Ingresar al sistema (acceso).
    - Ingresar registro de eventos (Registrar evento)
        - Riesgo Operativo
    - Ingresar registro de incidentes (Registrar incidente).
        - Incidente de seguridad
- Rol Supervisor
    - Ingresar al sistema (acceso).
    - Ingresar datos para el registro de eventos completo (Registrar evento)
        - Riesgo Operativo
        - Cuentas contables
        - Tipo evento
        - PAF
        - Canales
        - Proceso
        - Operación
        - Lugar

        - Líneas de negocio
- Rol Seguridad
    - Ingresar al sistema (acceso).
    - Ingresar datos para el registro de incidentes de seguridad completo (Registrar incidente)
        - Incidentes de Seguridad
- Rol Soporte
    - Ingresar al sistema (acceso).
    - Ingresar datos para el registro de incidentes de seguridad (Registrar incidente)
        - Incidentes de Seguridad
- Rol Administrador
    - Ingresar al sistema (acceso).
    - Ingresar datos para el registro de eventos completo (Registrar evento)
        - Riesgo Operativo
        - Cuentas contables
        - Tipo evento
        - PAF
        - Canales
        - Proceso
        - Operación
        - Lugar
        - Líneas de negocio
    - Ingresar datos para el registro de incidentes de seguridad completo (Registrar incidente)
    - Ingresar altas, bajas, modificaciones de usuarios.
    - Generación de copias de seguridad.

4.2. Procesos de Salida

- Rol Operador
    - Ver evento de riesgo (pdf, pantalla)
        - Riesgo Operativo
    - Ver incidente de seguridad (pdf, pantalla)
        - Incidente de Seguridad
- Rol Supervisor
    - Ver evento de riesgo (pdf, texto, pantalla)
        - Riesgo Operativo
        - Cuentas contables
        - Tipo evento
        - PAF
        - Canales
        - Proceso
        - Operación
        - Lugar

        - Líneas de negocio
- Rol Seguridad
    - Ver incidente de seguridad (pdf, pantalla)
        - Incidente de Seguridad
- Rol Soporte
    - Ver incidente de seguridad (pdf, pantalla)
        - Incidente de Seguridad
- Rol Administrador
    - Ver evento de riesgo (pdf, texto, pantalla)
        - Riesgo Operativo
        - Cuentas contables
        - Tipo evento
        - PAF
        - Canales
        - Proceso
        - Operación
        - Lugar
        - Líneas de negocio
    - Ver incidente de seguridad (pdf, pantalla)
        - Incidente de Seguridad
    - Ver reporte de los usuarios.
    - Ver las copias de seguridad.

5. REQUERIMIENTOS TÉCNICOS

5.1. Requerimientos mínimos de Hardware

Procesador: Core

Memoria RAM(Mínimo): 2 Gigabytes (GB)

Disco duro: 500 Gb.

5.2. Requerimientos mínimos de Software

Sistema Operativo: Windows 8, Windows 10 en Adelante.

Equipo computacional: PC, teclado, mouse, monitor, GB Tarjeta de red LAN y/o Wireless.

Conexión: internet local.

Navegador: Chrome, Mozilla, internet explorer.

6. HERRAMIENTAS USADAS PARA EL DESARROLLO

6.1. PHP

Es un Lenguaje de Programación para trabajar páginas WEB ofreciendo la ventaja de combinar con HTML. Las ejecuciones son realizadas en el Servidor y el cliente es el encargado

de recibir los resultados de la ejecución. Si el cliente realiza una petición, se ejecuta el intérprete de PHP y se genera el contenido de manera dinámica. Permite conexión con varios tipos de Bases de Datos como: MySql, SQL Server, etc. permitiendo aplicaciones robustas sobre la WEB. Este lenguaje de programación puede ser ejecutado en la gran mayoría de sistemas operacionales y puede interactuar con Servidores WEB populares

6.2. MySQL

Es un motor de Bases de Datos, el cual permite múltiples hilos y múltiples usuarios, fue desarrollado como software libre.

Aunque se puede usar sobre varias plataformas es muy utilizado sobre LINUX. Es libre para uso en Servidores WEB.

Ofrece ventajas tales como fácil adaptación a diferentes entornos de desarrollo, Interacción con Lenguajes de Programación como PHP, Java Script y fácil Integración con distintos sistemas operativos.

6.3. APACHE

Es un Servidor WEB desarrollado por el grupo Apache. Su código fuente se puede distribuir y utilizar de forma libre. Está disponible para diferentes plataformas de Sistemas Operativos entre otros Windows, Linux, Mac y NetWare.

Ofrece ventajas tales como independencia de plataforma, haciendo posible el cambio de plataforma en cualquier momento; creación de contenidos dinámicos, permitiendo crear sitios mediante lenguajes PHP. 3

Además de ser libre su soporte técnico es accesible ya que existe una comunidad que está disponible en foros, canales IRC y servidores de noticias, donde hay gran cantidad de usuarios disponibles para cuando surge algún problema.

6.4. XAMPP

XAMPP es un servidor independiente de Software libre, su nombre proviene del acrónimo X que significa que es multiplataforma y funciona en diferentes sistemas operativos como los Windows, Linux, Solaris y Mac OSX; la A es del servidor web apache, M de uso de MySQL (base de datos) y para los lenguajes de script PHP y Perl. El programa se distribuye bajo la licencia GNU (es la licencia más ampliamente usada en el mundo del Software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el Software) y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas; con lo cual será posible implementar la aplicación web y hacer las respectivas pruebas de su funcionamiento.

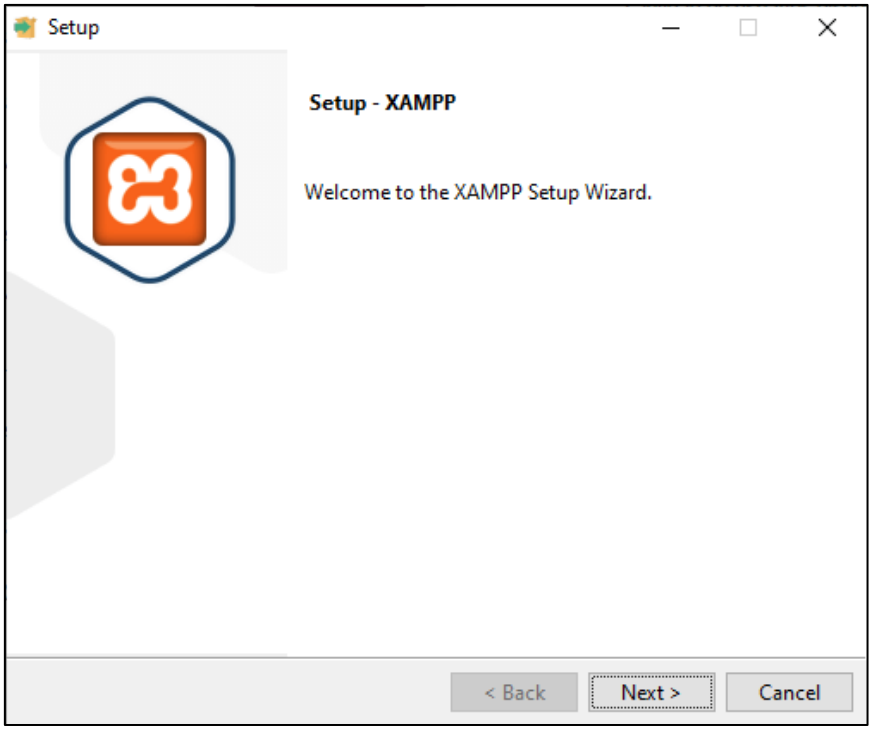
7. INSTALACIÓN DE WEB SERVER

Descargar en la PC el fichero de instalación xampp-windows-x64-8.1.10-0-VS16-installer Apache/2.4.34 (Win) 64b.exe el cuál se encuentra en la página: https://xampp.uptodown.com/windows una vez descargado, dar clic sobre él para iniciar la instalación. La instalación se debe hacer con una cuenta de Administrador o con derechos de administrador. Durante la instalación aparecerán las siguientes pantallas:

En la primera ventana aparece la pantalla de inicio del asistente para instalar XAMPP. Para ajustar las configuraciones de la instalación se hace clic en "Next".

Figura 1

XAMPP instalación



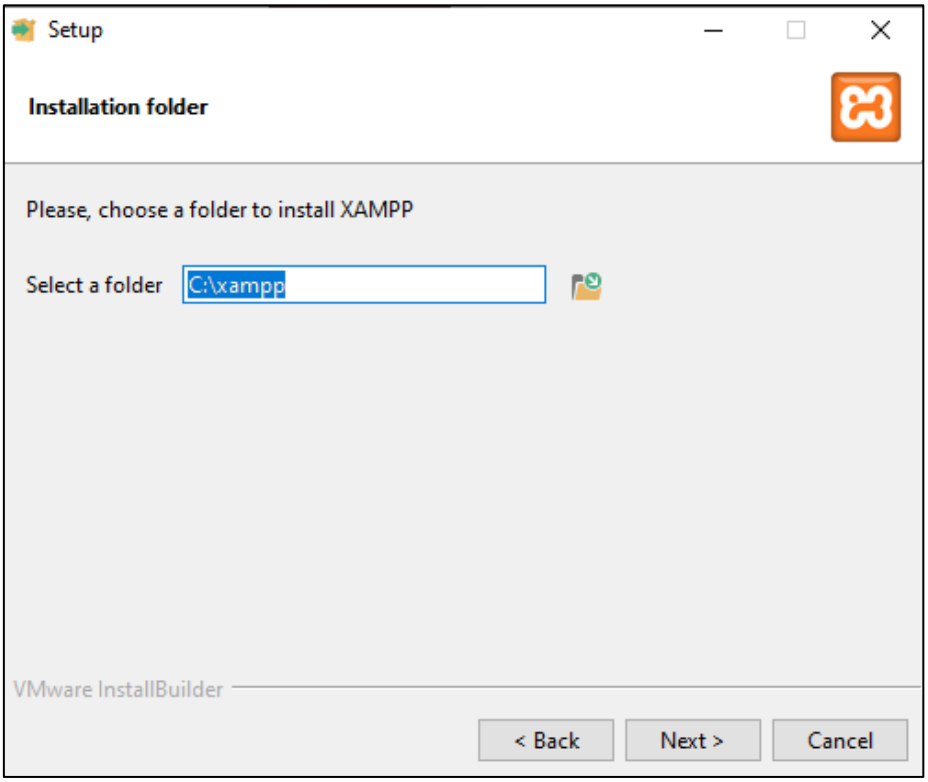
Nota. XAMPP Instalación (Elaboración propia)

La siguiente ventana presionar el botón Next Instalación y administración para continuar.

En este paso se escoge el directorio donde se instalará el paquete. Si se ha escogido la configuración estándar se creará una carpeta con el nombre XAMPP en C:\. Next Para continuar.

Figura 2

Direccionamiento para la instalación

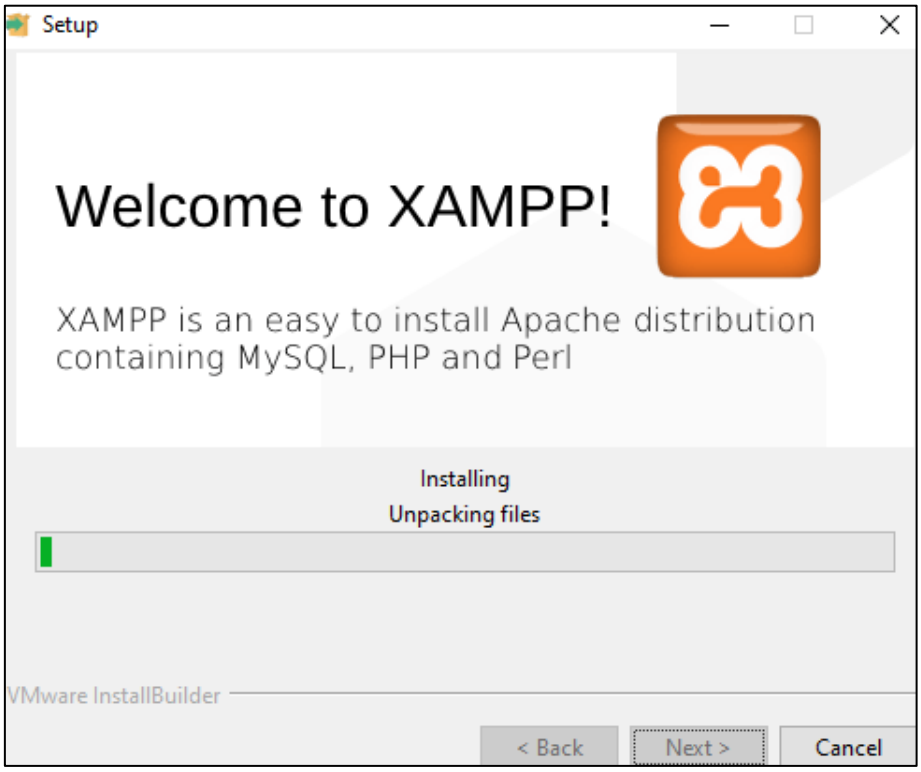


Nota. Direccionamiento para la Instalación (Elaboración propia)

El asistente extrae los componentes seleccionados y los guarda en el directorio seleccionado en un proceso que puede durar algunos minutos. El avance de la instalación se muestra como una barra de carga de color verde.

Figura 3

Iniciar Instalación

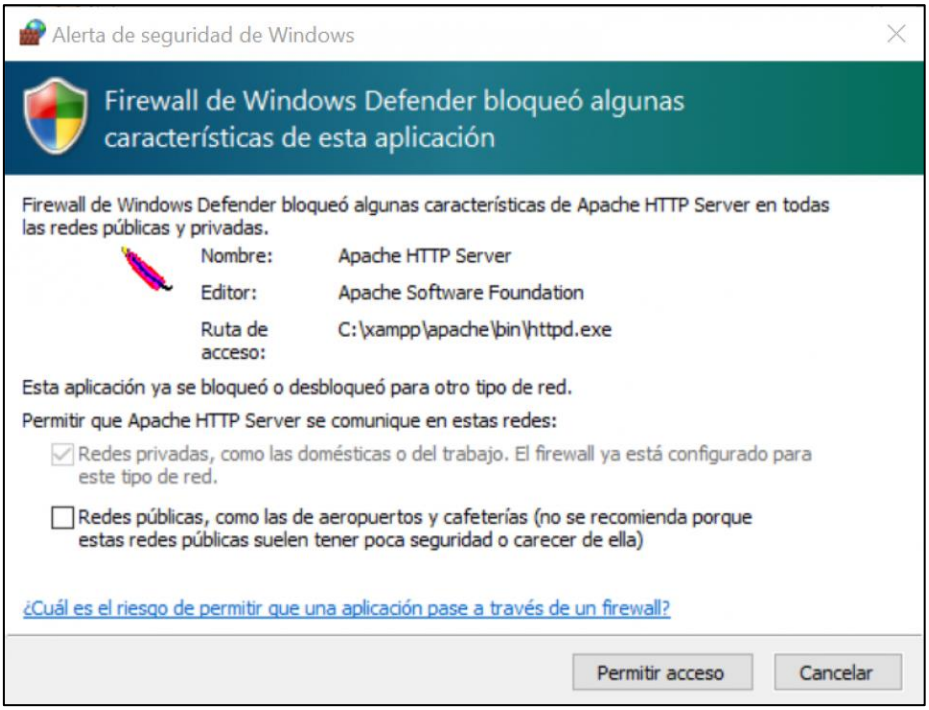


Nota. Iniciar instalación (Elaboración propia)

Durante el proceso de instalación es frecuente que el asistente avise del bloqueo de Firewall. En la ventana de diálogo puedes marcar las casillas correspondientes para permitir la comunicación del servidor Apache en una red privada o en una red de trabajo. Recuerda que no se recomienda usarlo en una red pública.

Figura 4

Configuración Firewall



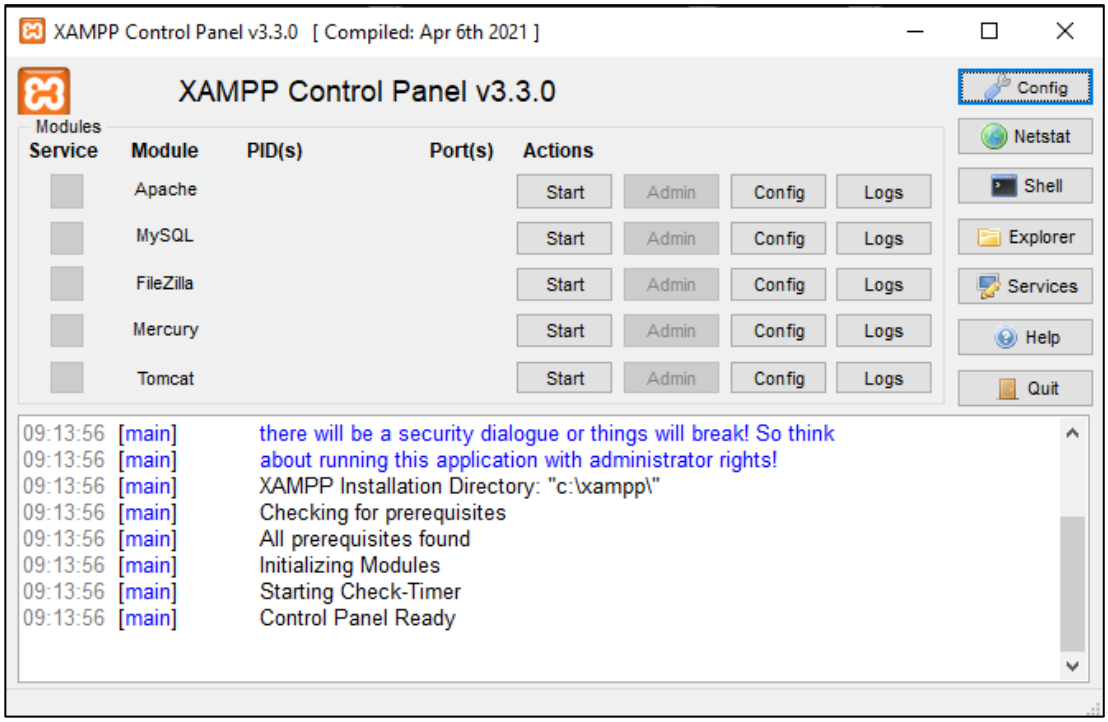
Nota. Configuración Fairewall (Elaboración propia)

Una vez extraídos e instalados todos los componentes puedes cerrar el asistente con la tecla "Finish". Para acceder inmediatamente al panel de control solo es necesario marcar la casilla que pregunta si deseamos hacerlo.

En la interfaz de usuario del panel de control se protocolan todas las acciones y es posible activar o desactivar los módulos por separado con un simple clic. Además, se dispone de diversas utilidades como:

Figura 5

Interfaz del servidor configurable

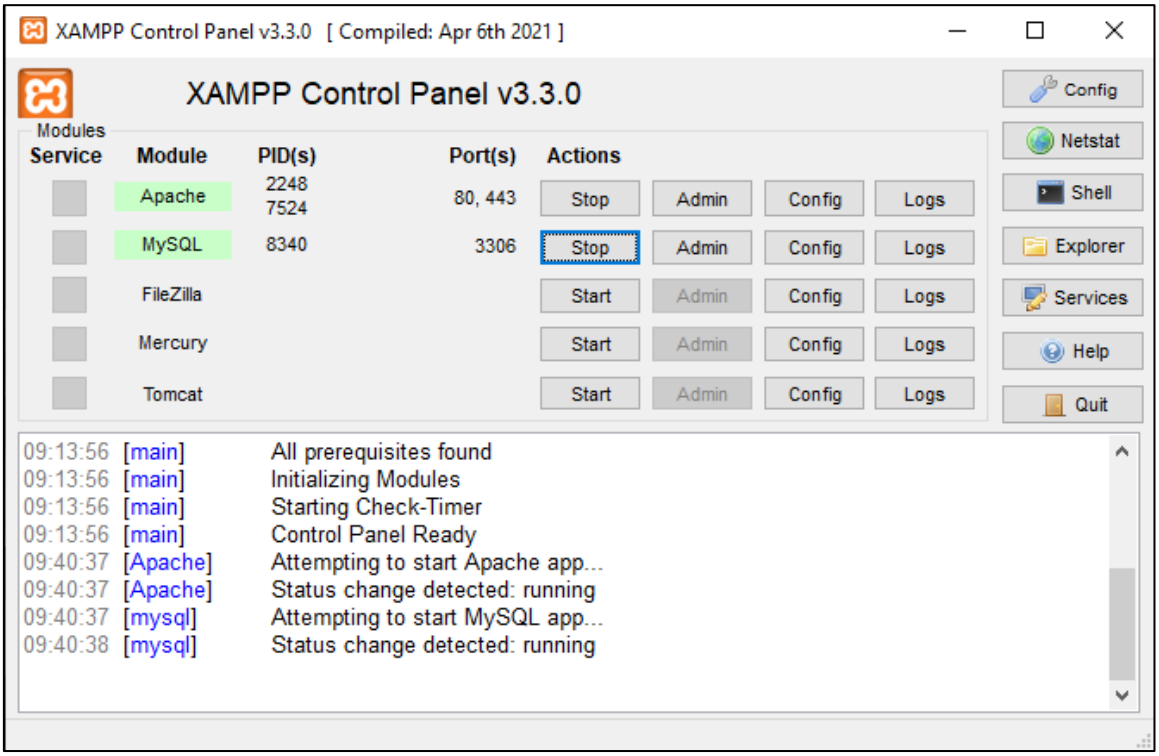


Nota. Interfaz del servidor configurable (Elaboración propia)

En la parte superior se pueden iniciar o interrumpir los módulos de XAMPP por separado mediante los comandos "Start" y "Stop" bajo "Actions". Los módulos que se activaron aparecen marcados en verde.

Figura 6

Iniciando el puerto Apache y MyQSL



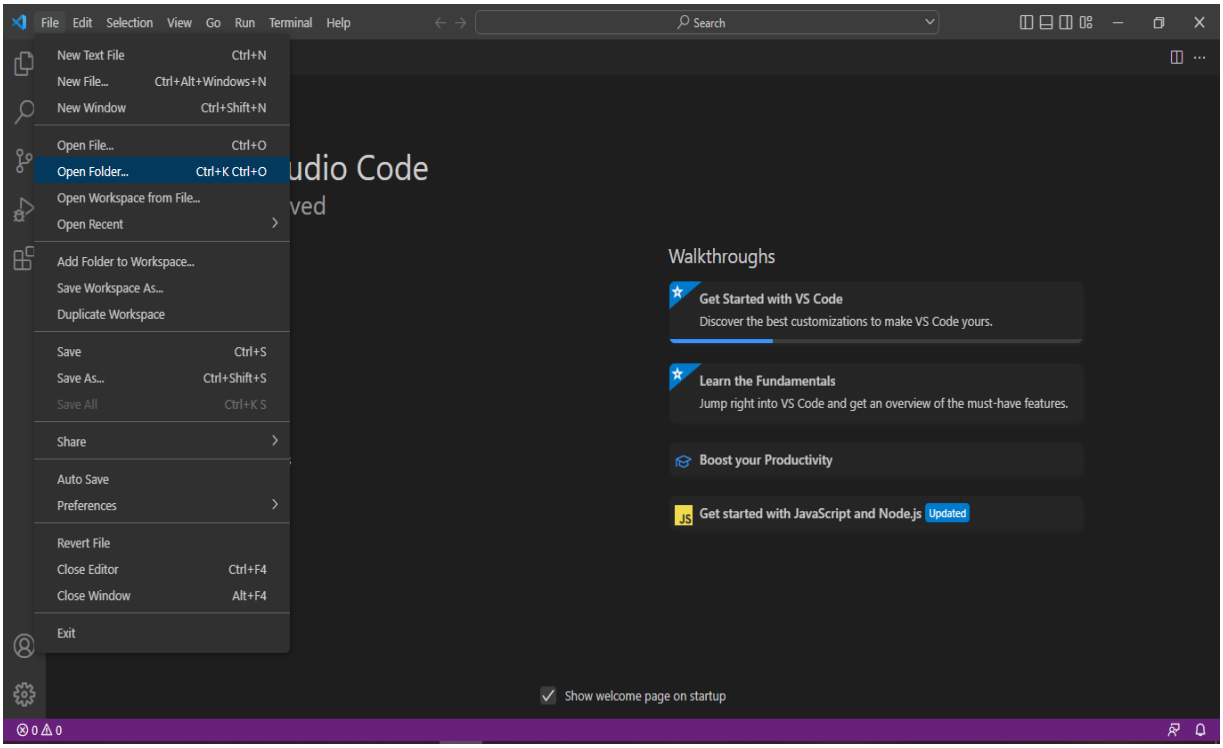
Nota. Iniciando el puerto Apache y MySQL (Elaboración propia)

8. CREANDO EL PROYECTO CIRO

Desde Visual Studio Code, damos clic en Flie, luego Open Folder y para finalizar nos direccionamos en la ruta donde se encuentra una carpeta llamada Ciro.

Figura 7

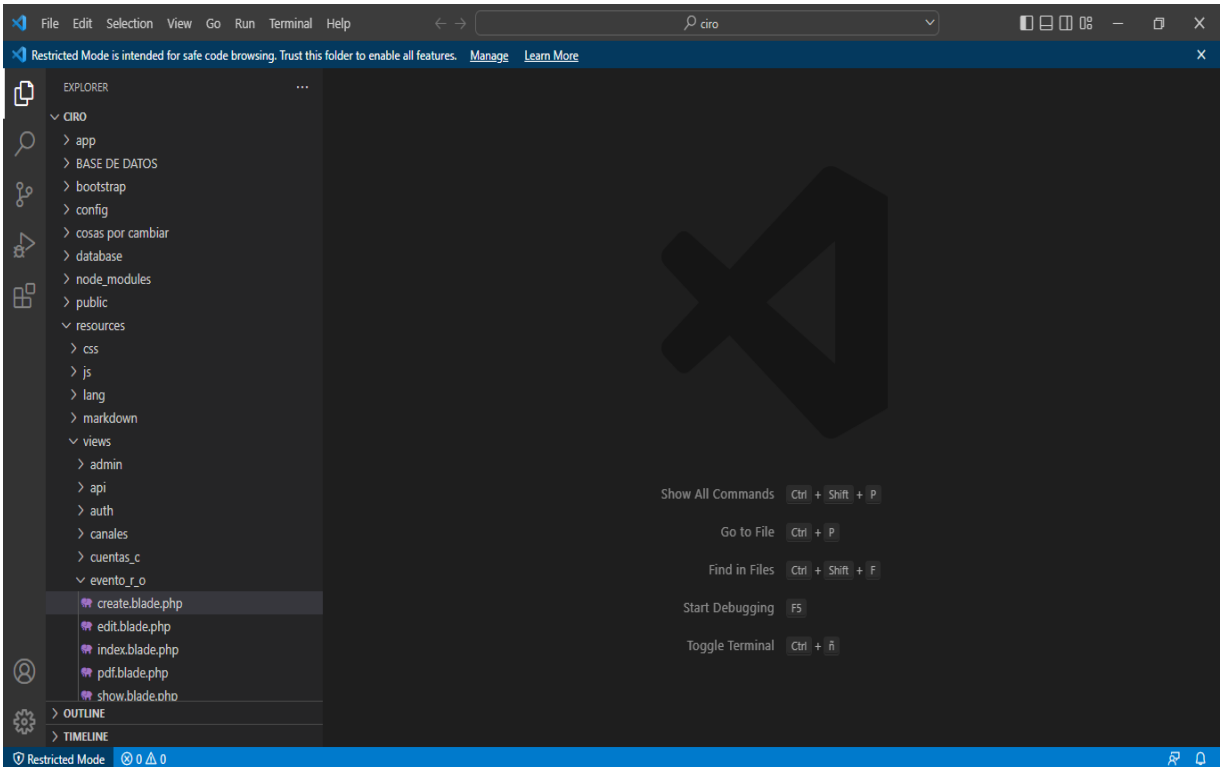
Direccionando el proyecto CIRO.



Nota. Direccionando el proyecto CIRO (Elaboración propia)

Figura 8

Apertura del proyecto CIRO.



Nota. Apertura del proyecto CIRO (Elaboración propia)

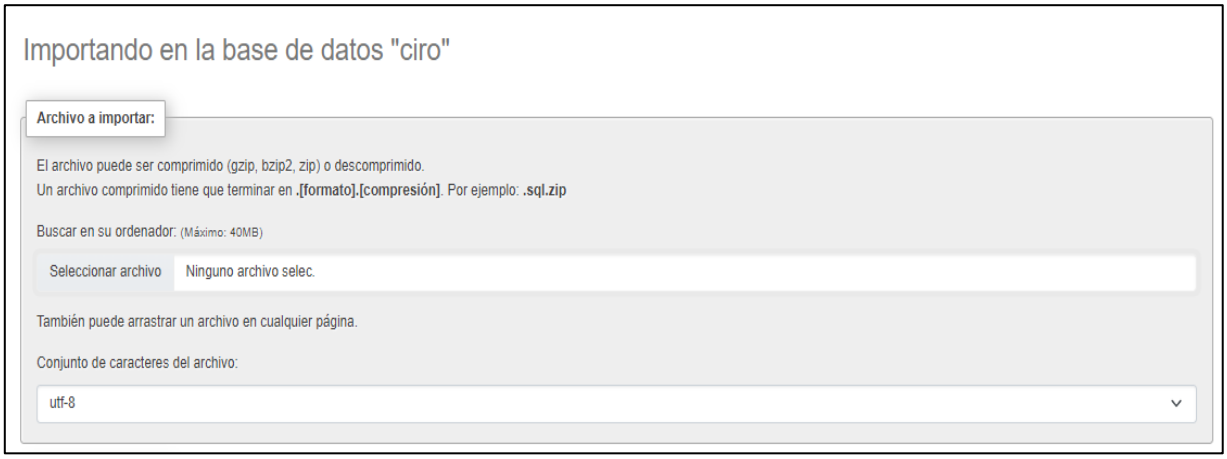
9. CREANDO EL PROYECTO CIRO EN EL GESTOR DE BASE DE DATOS

Desde la Base de Datos en la siguiente ruta http://localhost/phpmyadmin/index.php?route=/server/databases con el siguiente cotejamiento utf8mb4_general_ci y darle clic en el botón CREAR.

Figura 9

Creación de la Base de Datos "ciro"

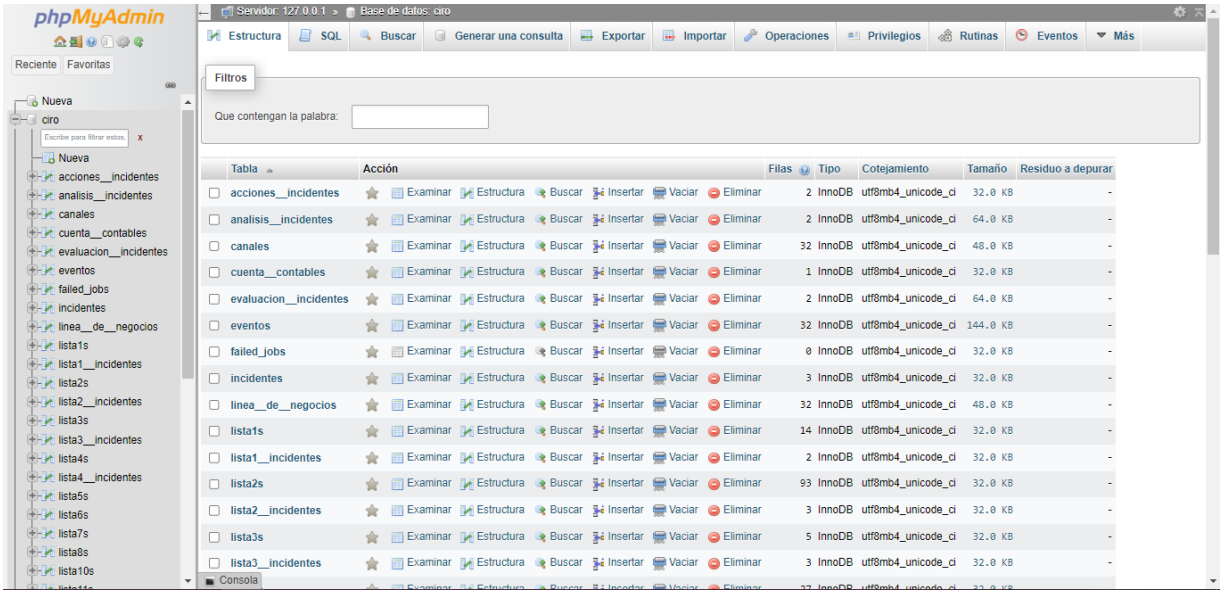
Nota. Creación de la Base de Datos "ciro" (Elaboración propia)



Luego se debe importar la base de datos llamada "ciro.sql"

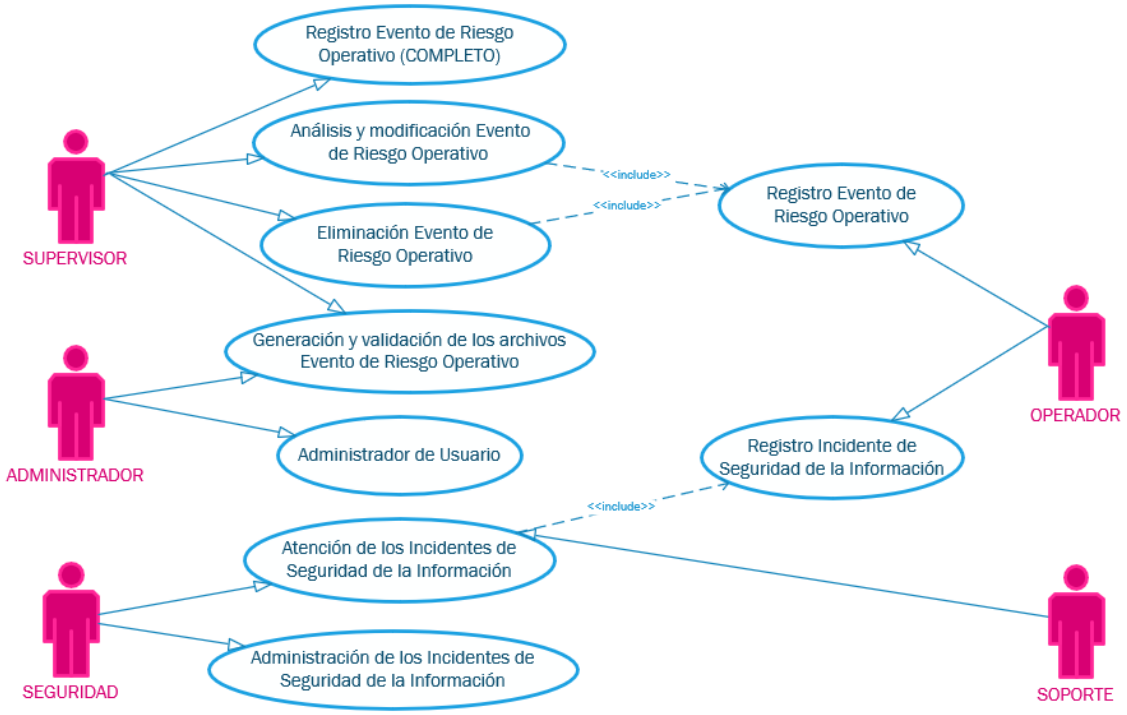
Figura 10

Base de Datos "ciro" importado correctamente



Nota. Base de Datos "ciro" importado correctamente (Elaboración propia)

10. DIAGRAMA DE CASO DE USO DE ALTO NIVEL

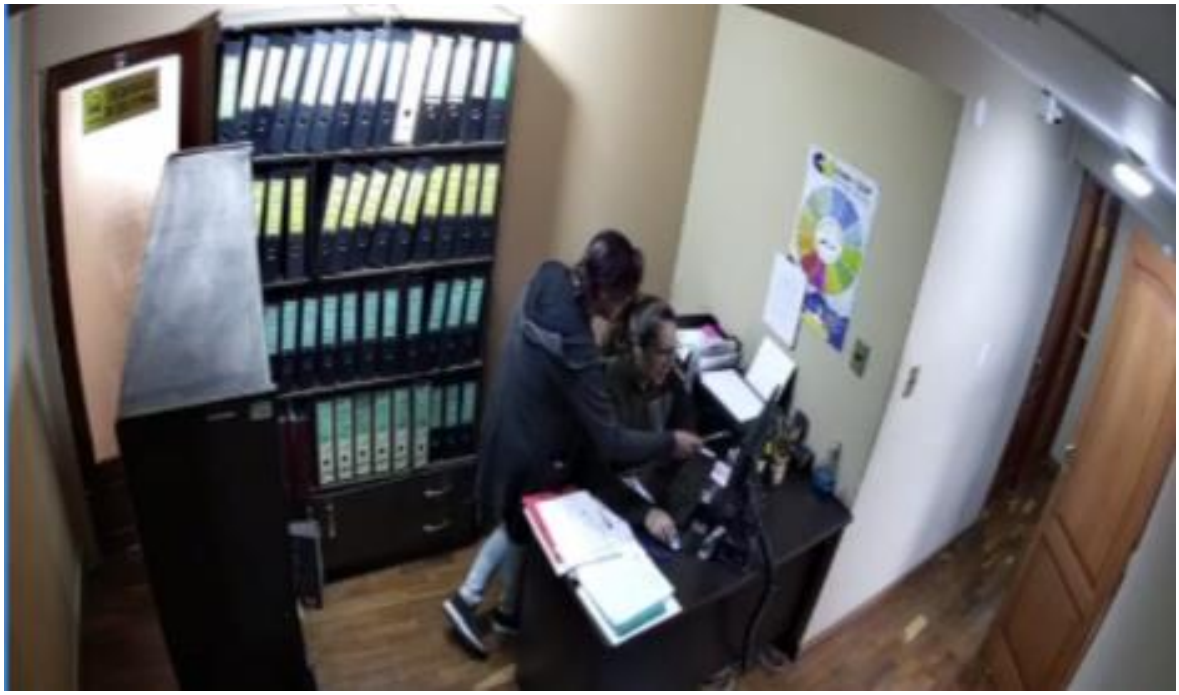


11. DIAGRAMA ENTIDAD RELACIÓN

ANEXO E

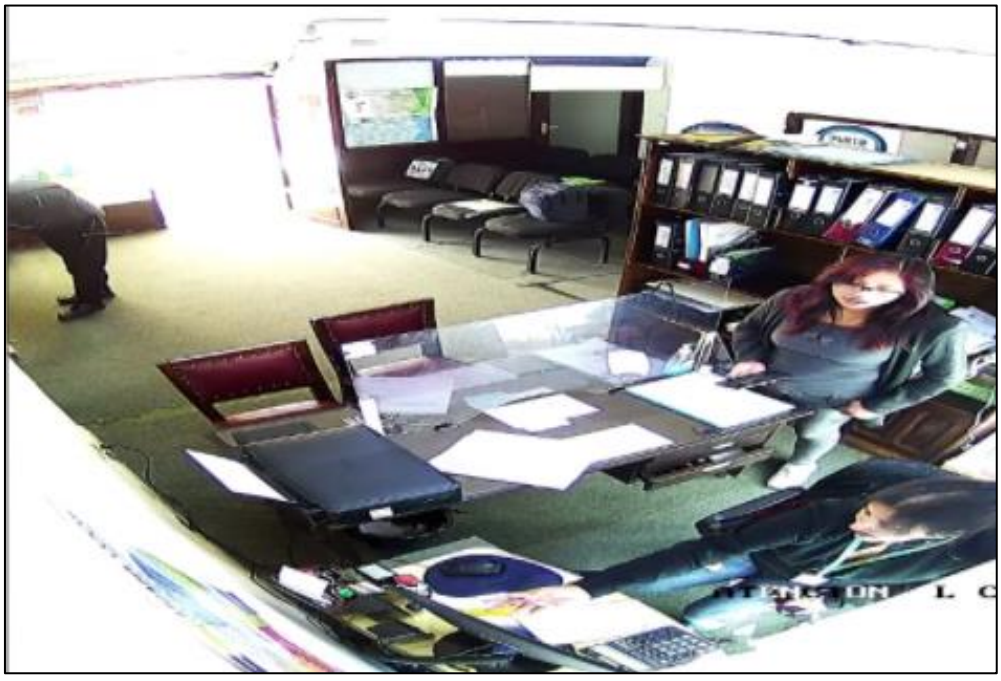
DOCUMENTOS DE RESPALDO

Día de capacitación el personal



Capacitando al personal en su área de trabajo

Capacitando al personal en su área de trabajo



Capacitando al personal en su área de trabajo

El Alto, Noviembre de 2022

Señor:
Ing. David Carlos Mamani Quispe
DIRECTOR DE LA CARRERA
INGENIERIA DE SISTEMAS
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido director de carrera:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad"
CASO: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. (PARA PROYECTO DE GRADO)
MODALIDAD: Proyecto de Grado
Univ.: Tania Ines Mamani Luque
Registro Universitario: 14007145
Cedula de Identidad: 7056228 LP.

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,

.....

Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II

El Alto, Noviembre de 2022

Señor:
Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

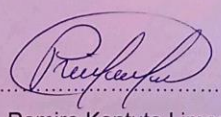
REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: "Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad"
CASO: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. (PARA PROYECTO DE GRADO)
MODALIDAD: Proyecto de Grado
Univ.: Tania Ines Mamani Luque
Registro Universitario: 14007145
Cedula de Identidad: 7056228 LP.

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,



Ing. Ramiro Kantuta Limachi
TUTOR REVISOR

El Alto, Noviembre de 2022

Señor:
Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: "Sistema de Información para el Seguimiento y Control del Riesgo Operativo e Incidentes de Seguridad"
CASO: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda. (PARA PROYECTO DE GRADO)
MODALIDAD: Proyecto de Grado
Univ.: Tania Ines Mamani Luque
Registro Universitario: 14007145
Cedula de Identidad: 7056228 LP.

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Atentamente,

.....
Ing. Freddy Salgueiro Trujillo
TUTOR ESPECIALISTA

COOPERATIVA DE AHORRO Y CRÉDITO
UNIÓN SANTIAGO DE MACHACA
"USAMA" LTDA.
Fund. el 19 de Julio de 1985 - Personeria Jurídica 06 de junio de 1994

El Alto, 18 de noviembre de 2022

Señor:
Ing. Marisol Arguedas Balladares
TUTOR METODOLOGICO – TALLER II
UNIVERSIDAD PUBLICA DE EL ALTO
Presente. -

Ref.: Aval de conformidad del Sistema de Información

Distinguida ingeniera.

Mediante la presente tengo a bien enviarle mis sinceros saludos y deseos de éxito en la labor que desempeña en tan prestigiosa casa superior de estudios.

El motivo por el cual me dirijo a su persona, es para **certificar** que la Srta. Tania Ines Mamani Luque con C.I.: 7056228 LP concluyó el **"Sistema de Información para el Riesgo Operativo e Incidentes de Seguridad" Caso: Cooperativa de Ahorro y Crédito Unión Santiago de Machaca USAMA Ltda.**, el cual se encuentra concluido e implementado satisfaciendo los requerimientos exigidos.

Sin otro en particular y agradeciendo su atención, me despido

Atentamente,

Lic. Marcelo Nestor Polo Vila
GERENTE GENERAL
COOPERATIVA DE AHORRO Y CREDITO
USAMA LTDA.

COOPERATIVA DE AHORRO Y CREDITO UNIÓN SANTIAGO DE MACHACA USAMA LTDA. La Paz - Bolivia

El Alto: Av. Juan Pablo II N°2805Zona Ferropetrol - Telf.: 2845771 - 77832237
La Paz:Calle Isaac Tamayo N°662, 2do Piso - Telf.: 2452970 - 79812776
Pagina Web: www.cooperativausama.com

CUESTIONARIO

Evaluación de calidad del sistema INFOSYS por los usuarios

NOMBRE: Teis Callisaya Gutierrez

CARGO: Técnico de Soporte e Infraestructura de Sistemas

Descripción del cuestionario:

El cuestionario consta de preguntas que especifican los requisitos que debe cumplir el sistema en su funcionamiento, para lo cual, requiere ser evaluada por los usuarios quienes serán los manipuladores del mismo.

Debe asignar una puntación a cada pregunta, calificando con una nota entre 0 a 5

Escala de valoración:

0	Ninguna
1	Insignificante
2	Moderada
3	Media
4	Significativa
5	Fuerte

1. ¿Requiere el sistema copias de seguridad y de recuperación fiable?

   0 1 2 3 4 (5)

2. ¿Se requiere comunicaciones de datos?

   0 1 2 3 (4) 5

3. ¿Existen funciones procesamientos distribuidos?

   0 1 2 (3) 4 5

4. ¿Es crítico el rendimiento?

   0 1 2 (3) 4 5

5. ¿Sera ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?

   0 1 2 3 (4) 5

6. ¿Requiere el sistema entrada de datos interactiva?

   0 1 2 3 (4) 5

7. ¿Facilidad operativa?

0 1 2 3 (4) 5

8. ¿Se actualizan los archivos maestros de forma interactiva?

0 1 2 (3) 4 5

9. ¿Son complejos las entradas, las salidas, los archivos o las peticiones?

0 1 2 (3) 4 5

10. ¿Es complejo el procedimiento interno?

0 1 2 3 (4) 5

11. ¿Se ha diseñado el código para ser reutilizable?

0 1 2 3 (4) 5

12. Facilidad de instalación

0 1 2 3 (4) 5

13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?

0 1 2 (3) 4 5

14. Facilidad de cambio

0 1 2 (3) 4 5

CUESTIONARIO

Evaluación de calidad del sistema INFOSYS por los usuarios

NOMBRE: Fanny Delia Nuncavi de Ramos

CARGO: Encargada de Captaciones y Tesorería

Descripción del cuestionario:

El cuestionario consta de preguntas que especifican la interactividad de uso del sistema, para lo cual, requiere ser evaluado por los usuarios quienes serán los manipuladores del mismo.

Debe encerrar en un círculo su respuesta **SI** o **NO**

1. ¿El sistema es fácil de utilizar?
   (SI) NO
2. ¿El sistema facilita el trabajo que usted realiza?
   (SI) NO
3. ¿Cómo considera los formularios que elabora el sistema?
   (SI) NO
4. ¿El sistema tiene la seguridad necesaria?
   (SI) NO
5. ¿Cómo considera el ingreso de datos del sistema?
   (SI) NO