Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

DISEÑO DE INVESTIGACIÓN PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA

Omar Francisco Mendoza Cajtic
Asesorado por el MSc. Ing. Edwin Estuardo Zapeta Gómez

Guatemala, julio de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DISEÑO DE LA INVESTIGACIÓN PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

OMAR FRANCISCO MENDOZA CAJTIC
ASESORADO POR EL MSC. ING. EDWIN ESTUARDO ZAPETA GÓMEZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, JULIO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. José Manuel Ruiz Juárez
EXAMINADOR	Ing. César Rolando Batz Saquimux
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE LA INVESTIGACIÓN PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha 7 de noviembre de 2020.

Omar Francisco Mendoza Cajtic

Escuela de Estudios de Postgrado, Edificio S-11
Teléfono: 2418-9142 / 24188000 ext. 1382
WhatsApp: 5746-9323
Email: informacion_eep@ing.usac.edu.gt
https://postgrado.ingenieria.usac.edu.gt

Ref. EEPFI-0004-2021
Guatemala, 18 de enero de 2021

Director
Carlos Gustavo Alonzo
Escuela de Ciencias y Sistemas
Presente.

Estimado Ing. Alonzo:

Reciba un cordial saludo de la Escuela de Estudios de Postgrado. El propósito de la presente es para informarle que se ha revisado y aprobado el DISEÑO DE INVESTIGACIÓN: **IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA**, presentado por el estudiante **Omar Francisco Mendoza Cajtic** carné número 200714554, quien optó por la modalidad del "PROCESO DE GRADUACIÓN DE LOS ESTUDIANTES DE LA FACULTAD DE INGENIERÍA OPCIÓN ESTUDIOS DE POSTGRADO". Previo a culminar sus estudios en la Maestría en Artes Tecnologías de la Información y la Comunicación.

Y habiendo cumplido y aprobado con los requisitos establecidos en el normativo de este Proceso de Graduación en el Punto 6.2, aprobado por la Junta Directiva de la Facultad de Ingeniería en el Punto Décimo, Inciso 10.2 del Acta 28-2011 de fecha 19 de septiembre de 2011, firmo y sello la presente para el trámite correspondiente de graduación de Pregrado.

Sin otro particular,

Atentamente,

"Id y Enseñad a Todos"

Ing. Estuardo Zapeta
Ingeniería en Ciencias y Sistemas
Colegiado 12767

UNIVERSIDAD DE SAN CARLOS
ESCUELA DE POSTGRADO
FACULTAD DE INGENIERIA
DE GUATEMALA

Mtro. Edwin Estuardo Zapeta Gómez
Asesor

Mtro. Marlon Antonio Pérez Türk
Coordinador de Área
Transferencia Tecnológica

FACULTAD DE INGENIERIA
DIRECCIÓN

Mtro. Ing. Edgar Darío Álvarez Cotí
Director
Escuela de Estudios de Postgrado
Facultad de Ingeniería

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÍA

EEP-EICS-001-2021

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador y Director de la Escuela de Estudios de Postgrado, del Diseño de Investigación en la modalidad Estudios de Pregrado y Postgrado titulado: **IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA**, presentado por el estudiante universitaria **Omar Francisco Mendoza Cajtic**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería en esta modalidad.

ID Y ENSEÑAD A TODOS

Digitally signed by Carlos Gustavo Alonzo
DN: 2.5.4.13=Profesional Titulado, c=GT, l=Guatemala / Guatemala, street=Via 5 3-65 zona 4 Ed. El Angel 5to nivel of 52, 2.5.4.20=22347420, ou=NA, o=NA, title=Ingeniero en Ciencias y Sistemas Colegiado. 6358, serialNumber=2278 03167 0101, 2.5.4.45=29020980, 2.5.4.27=06/03/79, email=carlosalonzo@infoutilitygt.com, cn=Carlos Gustavo Alonzo
Date: 2021.01.29 20:01:04 -06'00'

Ing. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, enero de 2021

Decanato
Facultad de Ingeniería
24189101- 24189102

DTG. 307.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado**: DISEÑO DE INVESTIGACIÓN PARA LA IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO EN TIEMPO DE EJECUCIÓN DE APLICACIONES QUE FUNCIONAN EN CANALES ELECTRÓNICOS DE UNA ENTIDAD BANCARIA PARA DETECCIÓN DE FALLAS DE FORMA PREVENTIVA,** presentado por el estudiante universitario: **Omar Francisco Mendoza Cajtic,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Anabela Cordova Estrada
Decana

Guatemala, julio de 2021

AACE/cc

ACTO QUE DEDICO A:

Dios	Una luz que siempre ilumina mi camino y guía cada uno de mis pasos en esta vida, a él sea toda la gloria, el poder y la honra.
Mis padres	Nicolas Mendoza y Azucena Cajtic, por el amor y apoyo incondicional que siempre encuentro en ellos.
Mi esposa	Mariela Castillo, por su comprensión, amor y apoyo en todo momento.
Mi hija	Monserrath Mendoza Castillo, un angelito lleno de inocencia para quien deseo ser un ejemplo.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	A la gloriosa Tricentenaria, por ser mi casa de estudios y permitir mi formación académica como profesional, de la cual estoy orgullosa de egresar.
Facultad de Ingeniería	Por todo el conocimiento y sabiduría adquiridos en sus aulas durante mis años de estudio.
Mis amigos	Personas que siempre me brindaron soporte en los momentos más difíciles de esta carrera.
Mi asesor	Msc. Ing. Estuardo Zapeta, por compartir su conocimiento y experiencia para la elaboración de este diseño de investigación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
$	Dólares
%	Porcentaje
Q	Quetzales

GLOSARIO

Almacenamiento de logs	Consiste en el empleo de un medio de almacenamiento permanente (archivos, bases de datos, etc.) para alojar logs generados y recolectados. Incluye la ejecución de tareas de rotación, archivado, comprensión, reducción, conversión, normalización y chequeo de integridad.
Análisis de logs	Se refiere al análisis del contenido de los logs con el objetivo de interpretarlo y obtener información relevante respecto a la administración de recursos, detección de intrusiones, resolución de problemas, análisis forense o auditorías dentro de la organización. Incluye la correlación de eventos, la visualización y exploración, y la generación de reportes.
Ciclo de vida de un log	Hace referencia a la serie de etapas de existencia del log en la organización desde su generación hasta su definitivo descarte y eliminación.
Eliminación de logs	Hace referencia a la eliminación de las entradas de logs almacenadas correspondientes a un criterio, que generalmente incluye una fecha y una hora determinadas.

Evento	Es una singular ocurrencia dentro de un ambiente, que involucra usualmente un intento de cambio de estado. Un evento incluye una noción de tiempo, la ocurrencia, y una descripción pertinente al evento o al ambiente que pueda ayudar a explicar o entender las causas o efectos del evento.
Fuente generadora de logs	Se refiere a todo dispositivo, sistema o aplicación que esté en la capacidad de registrar la ocurrencia de eventos.
Log	Es un registro de los eventos que ocurren dentro de los sistemas y redes de una organización.
Runbook	En un sistema informático o una red, un runbook es una compilación de procedimientos y operaciones de rutina que realiza el administrador o el operador del sistema.
Transporte y recolección de logs	Incluyen los diversos mecanismos disponibles para mover los logs desde la fuente generadora hacia una ubicación diferente, generalmente centralizada.

RESUMEN

La transformación digital y el aumento en la demanda de servicios en línea ha creado en las instituciones financieras la necesidad de crear una estrategia de omnicanalidad para mejorar la experiencia de los usuarios e impulsar mejores relaciones con los clientes a través de sus puntos de contacto especialmente en los canales electrónicos.

Esta demanda de servicios en línea ha obligado al sector financiero a reducir el tiempo comprendido desde que un producto o servicio es concebido hasta que está disponible para el usuario final. Esto implica que se deben optimizar los procesos en el desarrollo de la solución y eliminar aquellos que no aportan valor al objetivo de negocio que apoya el proyecto, con este fin se excluyen requisitos no funcionales durante la definición del proyecto para apresurar el inicio de su etapa de desarrollo.

Cuando la solución es implementada, estos requisitos que no se consideraron generan un comportamiento inesperado de las aplicaciones en producción. Estos fallos no son ocasionados por los equipos en los cuales se instala la solución, son provocados por el funcionamiento de la aplicación y esto no es detectado hasta que el fallo ha escalado de tal forma que ha afectado los recursos físicos del equipo donde se encuentra instalado y estos reportan una alerta de mal funcionamiento.

Este sistema de detección de fallas de forma reactiva permite que la aplicación siga funcionando aun cuando su comportamiento no es correcto y afecta otros sistemas que con los cuales comparte recursos físicos.

1. INTRODUCCIÓN

La pandemia que afecta el mundo en este 2020 ha sido catalogada por muchos expertos del área de informática y ciencias de la computación como el principal impulsor de un proceso de transformación digital para las organizaciones del mundo, y quienes se adaptaron a este cambio están experimentando una nueva forma de prestación de servicios por medio de canales digitales. Las empresas que iniciaron de forma acelerada la migración de sus procesos presenciales hacia los canales virtuales, están enfrentando algunos problemas derivados de la falta de planificación en la implementación de sus soluciones y el rápido escalamiento en el uso, por parte de los clientes, quienes cada día están más anuentes a usar canales digitales para evitar el contagio de COVID-19.

Entre las organizaciones que actualizaron su modelo de atención al cliente se encuentran las instituciones bancarias, un sector en el cual la interacción entre personas está muy arraigada en la cultura pero que ha tenido que cambiar y ahora se está dando un impulso al uso de los canales electrónicos para proteger la salud de los clientes y trabajadores. Este aumento en la demanda de servicios ha generado que los sistemas sean sometidos a cargas transaccionales muy altas que no estaban proyectadas y en consecuencia, los sistemas fallan.

Existen fallos de bajo impacto que el área de monitoreo, del departamento de mantenimiento de software, puede controlar y dar una solución rápida, pero existen otros más críticos que requieren mucho tiempo de investigación antes de encontrar una solución. Este estudio pretende implementar una solución a esta problemática, automatizando el monitoreo de aplicaciones que funcionan en

canales electrónicos de una entidad bancaria para detectar los fallos de forma preventiva y mejorar la estabilidad de los canales electrónicos.

El tema de estudio será la automatización de la detección de fallos a través de la gestión centralizada de logs, para obtener y consolidar datos del funcionamiento de las aplicaciones y sistemas que conforman el entorno para el funcionamiento de los canales electrónicos. Estos datos serán importados a través de la infraestructura informática de la institución bancaria hacia un almacén de datos en una ubicación central, con una estructura común y optimizada para su posterior lectura y análisis en condiciones adecuadas de rapidez y eficacia.

Se desarrollará un modelo de *machine learning* el cual buscará e identificara patrones entre los registros de eventos ubicados en una base de datos central, que estará siendo actualizada en tiempo real dando a este modelo la capacidad de detectar y predecir fallos en tiempo real de ejecución. Este sistema de monitoreo y notificación de alertas será una herramienta para que el área de mantenimiento de aplicaciones pueda realizar las tareas necesarias que garanticen el correcto funcionamiento del sistema en un momento oportuno.

Para la implementación de este sistema se desarrollará una arquitectura basada en plataformas de código abierto para cubrir las funciones de recuperación de logs, procesamiento de datos, transformación de mensajes, almacenamiento, búsqueda, y análisis de data. El estudio contempla una fase de experimentación para entrenar el modelo predictivo usando *machine learning* y que este sea capaz de dar los resultados esperados en la detección preventiva de fallas.

Para cubrir los aspectos técnicos, operativos y económicos de factibilidad del estudio se realizará una colaboración con una institución bancaria que está dispuesta a ofrecer su apoyo para la implementación exitosa del proyecto.

El trabajo de graduación constara de 6 capítulos, a continuación una breve reseña de cada uno:

- Antecedentes: aspectos que cambiaron la forma de usar los canales digitales y el impacto derivado de alta demanda de servicios a través de estos.

- Justificación: porque es importante monitorear nuestros sistemas y cómo impacta en el servicio al cliente la calidad de nuestros sistemas electrónicos.

- Alcances: definición de los alcances técnicos, investigativos y los resultados de este estudio.

- Marco teórico: una investigación documental sobre la gestión centralizada de logs y las herramientas que nos ayudaran a obtener resultados satisfactorios para este estudio.

- Presentación de resultados: esquema de los resultados obtenidos durante la fase de experimentación del estudio.

- Discusión de resultados: análisis de los resultados y sus implicaciones para el problema que se planteó.

2. ANTECEDENTES

La transformación digital impacto directamente en la vida de todas las personas, cambiando cómo se hacían las cosas antes. Este cambio no solo afecto a los usuarios de bienes y servicios, también afecto a las instituciones que brindan dichos bienes y servicios. Debido a estos cambios se adaptaron para hacer las cosas más rápidas y por medio de canales electrónicos. Pero, aunque la tecnología y el internet hicieron los procesos más rápidos y accesibles, existen problemas con los cuales las instituciones de distintos sectores deben enfrentar en sus operaciones diarias, como fallos en los sistemas o comportamientos anómalos de las aplicaciones, que impactan directamente en su servicio al usuario final.

Durante la operación del observatorio ALMA, Gil, Reveco y Shen (2016), diariamente se generan una gran cantidad de archivos de registro. Como el software ALMA todavía está en continua evolución, los registros no solo son útiles para diagnosticar fallas detectadas durante operación, pero también son necesarios para representar el análisis de rendimiento a largo plazo y proporciona una vista rápida del comportamiento sistema. Como cualquier otro software, el propósito del log es publicar cualquier tipo de estado e información de diagnóstico. Los logs son esenciales para el análisis postmortem de problemas de *hardware* o *software*, especialmente para un complejo sistema distribuido como el *software* de control ALMA.

Los fallos eran reportados como *tickets* y la resolución de estos implica investigaciones se llevan a cabo principalmente a través de la revisión de logs que proporciona información sobre el estado de los equipos y servidores de red

que podrían estar relacionados con el error informado. Dependiendo de la dificultad del problema y el nivel de experiencia del investigador en el área específica, una investigación podría demorar de 10 minutos a un par de días en ser resuelta.

Algunos inconvenientes derivados de esta metodología son la duplicidad de investigaciones que se realizadas para corregir un mismo problema, una alta generación de *tickets* porque se reporta el mismo fallo varias veces, dificultad para analizar toda la información almacenada en los *logs* debido a su gran tamaño y una falta de estadísticas de sobre la resolución de fallos a través del tiempo.

Para solucionar estos problemas optaron por la combinación de herramientas conocida como *ELK-Stack: Elasticsearch como base de datos de back-end, Logstash* como pipeline y formateador de registros, y Kibana como herramienta de visualización.

Lograron disminuir las investigaciones duplicadas al identificar el origen de fallos, esto también contribuyo a bajar el número de *tickets* reportados por el sistema. Usando ELK-Stack se proporciona una forma rápida y versátil de analizar toda información generada en *logs* lo que ayudo a bajar el tiempo invertido por los investigadores en encontrar el origen de fallos reportados. Usando los tableros generados por Kibana se implementó un control con estadísticas sobre la resolución de fallos a través del tiempo.

ELK-Stack como sistema de gestión de *logs* es muy útil en términos de usabilidad debido a que permite fácilmente realizar búsquedas sobre grandes cantidades de información y visualizar los resultados de forma ordenada. Una aplicación de esto puede ser la geo identificación de usuarios, Prakash, Kakkar y

Patel (2017) que acceden un sitio basándose en los *logs* de ingreso. Ellos han utilizado un pequeño conjunto de datos de registro sólo para demostrar la usabilidad de ELK-Stack para obtener información de un sistema y sus usuarios a través del análisis de información almacenada en los *logs*.

Otra implementación de ELK-Stack fue realizada en Brasil, Almeida *et. al.* (2017) con el objetivo de realizar observación meteorológica, utilizando para indexar datos y metadatos provenientes de los *logs* generados por casa uno de los sensores. Además de permitir la aplicación de la curva de calibración del sensor para corregir los datos medidos, el uso de dicha infraestructura también permitió el análisis de datos. Este caso es interesante porque nos muestra cómo se puede lograr una homologación de varios *logs* con formatos e información distinta y luego poder analizarla como un conjunto único de información consolidada.

A medida que se ha incrementado el uso de sistemas de administración de log, la industria del software ha adoptado una serie de criterios, Mitra y Sy (2016), sobre buenas prácticas para un correcto uso de LMS, entre ellos están:

- Definir una política de auditoría: se debe definir una amplia gama de tipos de eventos de seguridad que se pueden grabar en los registros.

- Consolidación de registros: permitir almacenar *logs* como registros de base de datos y archivo de texto plano comprimido para reducir el espacio de almacenamiento y permitir reimportar log de mucho tiempo atrás.

- Monitoreo de eventos: utilizar herramientas de monitoreo de log adicionales a la utilidad Windows Event Log.

- Generación de informes: los informes se deben proporcionar diaria, semanal, mensual y anualmente, y se debe tener la capacidad de generar reportes personalizados según la necesidad del cliente.

- Auditoria: la función de un LMS es convertir datos no estructurados en un conjunto de información útil que permite ser analizada e interpretada a través de búsquedas realizadas por parámetros.

Esta serie de buenas prácticas permiten que un sistema de monitoreo además de detectar fallas y comportamiento anómalo de aplicaciones también sea capaz de brindar una vista para evaluar el desempeño del sistema. La evaluación se realiza para verificar el rendimiento y el objetivo principal de la evaluación es asegurar que los componentes del sistema estén funcionando de forma correcta. Rochim, Aziz y Fauzi (2019).

Los tableros con graficas obtenidos de la ingesta y procesamiento de log son herramientas útiles que permiten visualizar como se está comportando el sistema y realizar una evaluación del rendimiento con mayor detalle y precisión, sobre todo en tiempo real y con datos de un ambiente de real.

Existen ciertos modelos de tableros que por la información que muestran siempre son implementados en un LMS. Prakash, Kakkar y Patel (2017), con algunos cambios según el giro de negocios de la institución. Las principales funciones de estos tableros son la visualización de *logs* recibidos por cada sistema, visualización por severidad, visualización de alertas de funcionamiento y visualización de estadísticas de rendimiento del sistema.

En conclusión, como se muestra a través de distintas implementaciones, ELK-Stack puede funcionar como una excelente herramienta para la gestión de

log, permitiendo consolidar información de distintas fuentes para que pueda ser analizada. Con esta herramienta se puede procesar toda la información generada en los *logs* de aplicaciones que funcionan en canales electrónicos de una institución bancaria para realizar un análisis de los fallos reportados y determinar su origen. Debido a que la información se almacena en una base de datos, también se pueden generar los tableros que se consideren necesarios para mostrar el funcionamiento del sistema en tiempo real.

3. PLANTEAMIENTO DEL PROBLEMA

La transformación digital y el aumento en la demanda de servicios en línea ha creado en las instituciones financieras la necesidad de crear una estrategia de omnicanalidad para mejorar la experiencia de los usuarios e impulsar mejores relaciones con los clientes a través de sus puntos de contacto especialmente en los canales electrónicos, los cuales debido a las restricciones de movilidad impuestas por el Gobierno de Guatemala para disminuir los contagios de COVID-19, han incrementado su demanda de servicio.

Esta demanda de servicios en línea ha obligado al sector financiero a reducir el tiempo comprendido desde que un producto o servicio es concebido hasta que está disponible para el usuario final. Esto implica que se deben optimizar los procesos en el desarrollo de la solución y eliminar aquellos que no aportan valor al objetivo de negocio que apoya el proyecto, con este fin se excluyen requisitos no funcionales durante la definición del proyecto para apresurar el inicio de su etapa de desarrollo. Cuando la solución es implementada, estos requisitos que no se consideraron generan un comportamiento inesperado de las aplicaciones en producción.

Estos fallos no son ocasionados por los equipos en los cuales se instala la solución, son provocados por el funcionamiento de la aplicación y esto no es detectado hasta que el fallo ha escalado de tal forma que ha afectado los recursos físicos del equipo donde se encuentra instalado y estos reportan una alerta de mal funcionamiento. Este sistema de detección de fallas de forma reactiva permite que la aplicación siga funcionando aun cuando su comportamiento no es correcto y afecta otros sistemas con los cuales comparte recursos físicos.

La institución financiera objeto de estudio tiene operación en Guatemala y Honduras, con una red de más de 1000 agencias, más 2000 agentes bancarios y canales de atención electrónica como kiosco virtual, banca virtual, aplicación móvil y asistente virtual, y cuenta con su propio Departamento de Tecnología. Cuando el área de mantenimiento y monitoreo de sistemas detecta una alerta, los administradores de los equipos físicos y virtuales inician una tarea de detección del origen de la falla enfocada en los aspectos técnicos que reportaron según sus protocolos de servicio y severidad de la alerta.

Debido a que las bitácoras únicamente guardan datos sobre los recursos físicos, solamente se puede determinar cuál fue el recurso afectado, por ejemplo, disco duro, memoria RAM, unidad de procesamiento, tráfico de red, pero no es posible determinar cuál fue el origen del problema y la aplicación que origino dicha falla, por lo que cuando esta se repita no será posible identificarla y únicamente se procederá a realizar el protocolo para detección de fallas reactiva documentado por la unidad de mantenimiento y monitoreo.

Si se revisa la documentación del proyecto con especificaciones de la solución no se define una sección de requisitos no funcionales que indique dominio de usuario y de datos que deberá soportar la aplicación para establecer límites óptimos de funcionamiento e instalar la aplicación en un equipo que soporte dichos límites.

Adicional, se determinó que existen fallas que son originadas por deficiencias en el código, y aunque existen herramientas de revisión de código estático como StyleCop, Gendarme, Checkstyle y FindBugs. Novak, Krajnc y Zontar (2010) que permiten aumentar su eficiencia, estas no son incluidas en el proceso de desarrollo de software por distintos motivos, principalmente porque

muchas de las aplicaciones están ligadas al core bancario basado en *Visual Basic* 5 un lenguaje que no tienes soporte técnico desde hace muchos años.

Existen otros inconvenientes, por ejemplo, muchas funcionalidades son programadas en procedimientos almacenados en la base de datos, el repositorio central no se encuentra actualizado y no existe un set de pruebas automatizado porque el lenguaje origen no lo permite. Es importante mencionar, que para las nuevas aplicaciones se tiene una estrategia para eliminar estos problemas porque están desarrolladas utilizando tecnología que si permite la integración de estas herramientas y minimizar las fallas en las aplicaciones originadas por deficiencias de código.

Debido a la transaccionalidad que manejan las operaciones bancarias siempre se deben hacer peticiones a la base de datos por lo que el fallo de una aplicación afecta directamente el rendimiento de esta a nivel general. Esta degradación en el rendimiento de la base de datos genera que las peticiones no sean atendidas de forma correcta generando una saturación de conexiones que generalmente se resuelve utilizando un protocolo documentado del área de monitoreo que consiste en eliminar todas las peticiones activas, creando inconsistencia en los datos de las peticiones truncadas.

Debido a que las soluciones están instaladas en canales electrónicos también afectan el tráfico en la red interna de la institución, esto genera una degradación del servicio para otras aplicaciones generando retraso o pérdida de solicitudes.

Todos estos inconvenientes generan un incremento en las solicitudes en el centro de ayuda, tanto de primera línea que son las que ingresan los clientes,

como las de segunda línea que se ingresan para soporte técnico por los agentes del centro de soporte.

Determinar de forma preventiva todos estos fallos disminuiría el número de quejas o solicitudes en centro de ayuda y determinar el origen de la falla es crucial para realizar una corrección que optimice el funcionamiento de la aplicación.

Cada software siempre tiene datos de registro, esto son esenciales para los desarrolladores y usuarios finales. Con los datos de registro se puede identificar la información de error del sistema, eventos de advertencia y si los procesos del sistema se están cargando con éxito.

Yang, Kristiani, Wang y Liu (2019), desarrollo un sistema de monitoreo para los datos de registro para realizar un análisis más detallado del funcionamiento de su sistema utilizando el sistema de almacenamiento Ceph y la tecnología *Elastic Stack*.

Ceph proporciona una gran cantidad de almacenamiento escalable y ELK consta de tres componentes: *Elasticsearch*, *Logstash* y *Kibana*. *Elasticsearch* es un motor de búsqueda para todo tipo de documentos, *Logstash* se usa para administrar centralizar, transformar y guardar los datos en *Elasticsearch* y *Kibana* se usa para visualizar datos de *Elasticsearch*, es una excelente herramienta para el análisis de datos en tiempo real y la toma de decisiones.

A partir de lo explicado anteriormente se plantea la siguiente pregunta central de investigación:

- ¿Cómo implementar un control de fallos preventivo para las aplicaciones que funcionan en canales electrónicos de una entidad bancaria?

Derivado de la pregunta central surgen las siguientes preguntas auxiliares:

- ¿Cómo se puede utilizar *Elastic Stack* para monitoreo de funcionamiento de aplicaciones de una entidad bancaria?

- ¿Cómo identificar fallos de operación en una aplicación que funciona en canales electrónicos de una entidad bancaria por medio de algoritmos de *machine learning*?

- ¿Cómo verificar la eficacia del sistema de monitoreo para aplicaciones que funcionan en canales electrónicos de una entidad bancaria?

4. JUSTIFICACIÓN

La línea de investigación sobre la cual se desarrolla este trabajo es desarrollo de sistemas para complementar la tecnología móvil.

Las medidas de distanciamiento social requeridas para la contención de la pandemia generada por el COVID-19 y las restricciones de movilidad impuestas por el Gobierno de Guatemala han generado en el país un incremento en la demanda de servicios electrónicos.

Empresas que contaban con este tipo de canales como una alternativa para sus usuarios se han visto afectadas por este incremento en el tráfico de solicitudes. Las entidades bancarias son un sector que ahora se enfrentan a este problema debido a que los productos y servicios ofrecidos en sus canales electrónicos era solo una parte de su portafolio y la otra parte era administrada en sus agencias por el modo tradicional de persona a persona.

Para adaptarse a esta nueva normalidad y mantener sus operaciones con regularidad se implementaron productos y servicios a través de sus canales electrónicos con la mayor premura posible que minimizara el impacto financiero que implicaba la denegación de servicios al no poder operar desde sus agencias físicas.

Estos servicios presentan fallos derivados del corto tiempo de desarrollo que impactan directamente en la calidad del servicio al cliente, con la presente investigación se pretende implementar una forma de identificar estos fallos estableciendo un sistema de monitoreo en tiempo de ejecución de aplicaciones

que funcionan en canales electrónicos de una entidad bancaria para detección de fallas de forma preventiva.

Elastic Stack es un grupo de productos de código abierto de *Elastic* diseñado para ayudar a los usuarios a tomar datos de cualquier tipo de fuente y en cualquier formato y buscar, analizar y visualizar esos datos en tiempo real.

Implementar *Elastic Stack* para la centralización de log de sistemas y aplicaciones que funcionan en canales electrónicos de una entidad bancaria, proporcionaría una herramienta para realizar un análisis detallado del funcionamiento de dichas aplicaciones lo que permitiría detectar fallos de forma preventiva mejorando la eficacia de los programas impactado directamente y de forma positiva en el servicio entregado al cliente final.

5. OBJETIVOS

5.1. Objetivo general

Implementar un sistema de monitoreo en tiempo de ejecución de aplicaciones que funcionan en canales electrónicos de una entidad bancaria para detección de fallas de forma preventiva utilizando algoritmos de *machine learning*.

5.2. Objetivos específicos

- Implementar *Elastic Stack* para la centralización de *logs* de sistemas operativos y log de aplicaciones que almacenan información de cada ejecución, para el análisis de datos del sistema en tiempo real.

- Identificar y predecir fallos de operación en una aplicación que funciona en canales electrónicos de una entidad bancaria utilizando algoritmos de *machine learning*.

- Realizar un tablero que muestre estadísticas iniciales de reportes de fallos que se actualice con los datos del sistema de monitoreo, para verificar la eficacia del sistema de monitoreo para aplicaciones que funcionan en canales electrónicos de una entidad bancaria.

6. NECESIDADES POR CUBRIR Y ESQUEMA DE SOLUCIÓN

6.1. Necesidades por cubrir

La transformación digital en el sector bancario ha favorecido la aparición de nuevas capacidades para utilizar la información y datos a través de canales electrónicos. Esta nueva forma de relacionarse con los clientes se ha dado a través de un corto periodo de tiempo, en una época en la cual el tiempo que tarda un producto en ser concebido hasta que está disponible para la venta es sumamente importante para ganar a la competencia.

Este proceso ha derivado en la instalación de soluciones de *software* que mejoren los procesos operativos y administrativos realzando todo aquello que realmente agrega valor al objetivo de negocio y fortalece la relación con el cliente. Sin embargo, estas soluciones son diseñadas y desarrolladas en un corto tiempo, y su implementación no está sujeta a un plan de monitoreo para garantizar la estabilidad del sistema.

Esta falta de monitoreo genera que las aplicaciones puedan tener fallos o comportamientos anómalos que no pueden ser detectados hasta que escalan al recurso físico en cual están instaladas generando una degradación en cascada del sistema a nivel general. La presente solución pretende contribuir con la creación de un sistema de monitoreo que satisfaga la necesidad de identificar fallos de forma reactiva y tener una visión del comportamiento del sistema en tiempo real.

6.2. Esquema de la solución

Se pretende implementar un sistema de monitoreo en tiempo de ejecución, para aplicaciones que funcionan en canales electrónicos. Este sistema permitiría la detección de fallas de forma preventiva evitando que el comportamiento anómalo de una aplicación escale a niveles que puedan afectar el servidor físico donde se encuentre instalado o la calidad de servicio al cliente.

Se utilizará *Elastic Stack* para procesar y analizar la información de cada log generado por las aplicaciones que formen parte del sistema. Con este objetivo, se definirá un log que guardé información de cada ejecución para que el sistema de monitoreo evalué su comportamiento en tiempo real. La definición de métricas de operación para cada aplicación que se incluya en el sistema de monitoreo a implementar es importante para tener un parámetro de configuración que determine el correcto funcionamiento.

Otra característica importante de esta herramienta se deriva de la persistencia de los datos de log en la base de datos *Elasticsearch*, lo que permite la generación de reportes en lapsos de tiempo que permiten evaluar el estado tanto del sistema como de las aplicaciones y establecer un control de fallos reportados y su recurrencia a través de periodos de tiempo constante, para visualizar como estos incidentes se comportan a través del tiempo antes y durante la instalación del sistema de monitoreo.

Los diferentes componentes de *ELK Stack* proporcionan una solución simple pero potente para la gestión de registros y análisis. Los diversos componentes en *ELK Stack* fueron diseñados para interactuar y funcionar bien entre ellos sin demasiada configuración adicional. Para la solución se propone el siguiente diseño de la pila de componentes, en la figura 1:

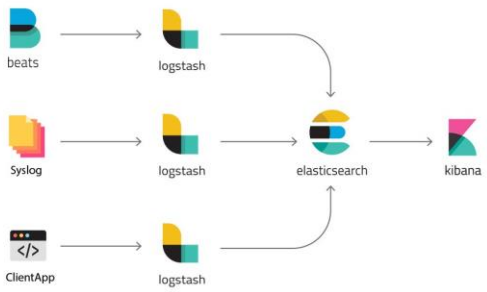
Figura 1. **Pila de componentes**

Fuente: elaboración propia.

Como se muestra en la figura 1, el componente *Logstash* realiza el procesamiento y carga en tiempo real de los *logs* de cada aplicación que forme parte del sistema de monitoreo, esta información será almacenada en el a base de datos *Elasticsearch* y para acceder y visualizarla se utilizara *Kibana*.

La en la figura 2, muestra cómo se procesa la información generada por los *logs*, a través de *Logstash* se realizará la ingesta de distintas fuentes del sistema, estos datos serán almacenados en la base de datos de *Elasticsearch* y se utilizará *Kibana* para acceder a ellos.

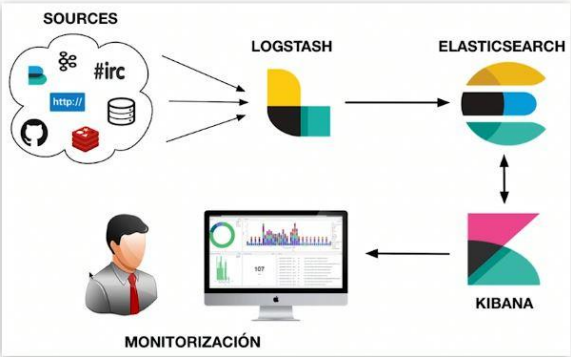
Figura 2. **Esquema de procesamiento de *logs***



Fuente: elaboración propia.

Para la función de monitoreo y detección de fallo se definirán tableros en *Kibana*, observando el rendimiento y comportamiento del sistema en tiempo real. La figura 3 muestra el proceso para consolidar varias fuentes de información, para posteriormente visualizar el resultado del análisis realizado, por medio de un tablero definido en *Kibana* para su monitorización.

Figura 3. **Esquema de monitoreo**



Fuente: elaboración propia.

Para detectar fallas de forma preventiva se desarrolla un algoritmo de análisis de toda la data almacenada en *Elasticsearch*, para determinar los patrones previos a cada falla. Como el ingreso de data de las aplicaciones se realiza en tiempo real de ejecución, cuando un patrón coincida se generará una alerta de una posible falla, indicando la aplicación y el comportamiento que podría resultar si no se corrige la falla. La criticidad de las alertas se puede parametrizar según un nivel de concordancia con el patrón establecido de la falla, es decir, entre mayor similitud se asigna una criticidad de atención más alta. Se pretende establecer un semáforo de alertas, con las siguientes características: Amarillo-Alerta de precaución, Anaranjado-Alerta de riesgo de fallo y Rojo-Alerta de falla inminente del proceso.

7. MARCO TEORICO

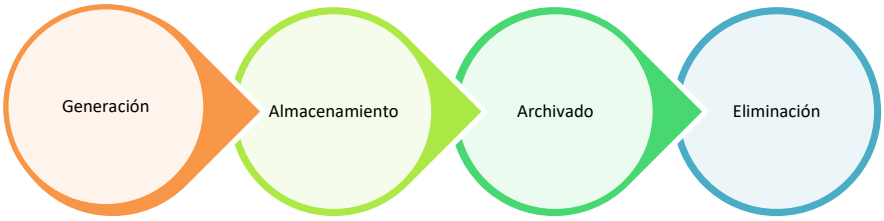
7.1. Gestión de logs

En entornos que trabajan con tecnología como una herramienta en su giro de negocios existen muchas aplicaciones que funcionan como fuentes de generación de *logs*, usualmente cada log posee un formato distinto y se almacena en distintos sitios, según la aplicación que lo genere. Para obtener información de valor para el negocio de esta gran cantidad de datos sin estructura, es necesario recopilar, filtrar y estructurar cada uno de los mensajes de las diversas fuentes y almacenarlos de forma centralizada, esto permite acceder fácilmente a ellos para su revisión o archivado para fines de auditoria. Para realizar la implementación de un sistema de gestión de log es necesario conocer los principios básicos descritos a continuación. West-Brown, *et al.* (1998).

7.1.1. Fases de un archivo de registros

Un log es un registro de eventos generados por un sistema y posee una serie de fases descritas en la figura 4. El registro se inicia en la ejecución de una aplicación, y se almacena durante el tiempo que esta esté operando, en el mismo sitio de almacenamiento. Cuando la ejecución termina, los *logs* son archivados en sitios distintos y se mantienen ahí por un periodo de tiempo, previo a ser eliminados Chuvakin y Schmidt (2012).

Figura 4. **Fases de un archivo de registros**



Fuente: elaboración propia.

7.1.1.1. Generación de *logs*

Si se administra un entorno con varios sistemas digitales, la mayoría de ellos pueden ser configurados para crear archivos de registros de eventos Kent y Souppaya (2006). Estos archivos pueden ser divididos según su naturaleza:

- Seguridad

Se usan principalmente para detectar y responder a ataques, infecciones de códigos maliciosos, suplantación de identidad y otros incidentes de seguridad.

- Operaciones

Se crean para guardar información de eventos en la ejecución de tareas y procesos del sistema.

- Depuración de aplicaciones

Los programadores utilizan este tipo específico de registro en el entorno de desarrollo. No se implementan en producción debido a impactos en el rendimiento, pero se pueden activar por medio de banderas para determinadas aplicaciones si fuera necesario.

Según Chuvakin y Schmidt (2012) existen tres componentes que se deben tomar en cuenta, durante el proceso de generación de registros de eventos:

- Transporte

Establece la forma en la cual se realizará el movimiento de archivos entre distintas ubicaciones. Por ejemplos, usando protocolos UDP, TCP, SOAP sobre HTTP y SNMP.

- Sintaxis

Establece la estructura interna del archivo de registros. cómo se forma el registro, cómo guardarlo y cómo lo visualiza el usuario para su análisis. Existen algunas estructuras usadas dentro de la industria del *software* que por su eficacia se han consolidado como estándar, tal es el caso de Syslog, Apache o el utilizado por CISCO.

- Formato del log

En este se establece cual es tipo de información que se almacenara. Se debe consignar data que genere un valor agregado para cada aplicación y por esto no hay un estándar.

También se debe considerar que existen muchos tipos de eventos que pueden ser almacenados y estos pueden ser categorizados Chuvakin y Schmidt (2012), aunque no es un estándar, como lo muestra la tabla I:

Tabla I. **Operaciones registradas en los logs**

Tipo de evento	Descripción del tipo de evento
Cambios administrativos en el sistema	Guarda cambios en el sistema, componentes, cuentas de usuario y actualizaciones.
Autenticación y autorización	Documente las decisiones e intentos de autenticación y autorización para usuarios.
Administración de permisos	Eventos relacionados a la política de roles y autorizaciones por usuario.
Gestión de amenazas.	Eventos de vulnerabilidad de la seguridad de sistemas e información.
Gestión recursos	Registro de eventos relacionados al rendimiento y estado de los recursos físicos de los sistemas.
Disponibilidad y continuidad del negocio	Archivo del estado de un sistema en un momento determinado.
Errores y fallas	Comportamiento no esperado del sistema o aplicación
Tramas de depuración	Mensajes asociados a peticiones que se desean monitorear.

Fuente: elaboración propia.

7.1.2. Conservación de archivos de registro

Dado que el análisis de eventos requiere la disponibilidad de todos los datos recolectados, usualmente los archivos se mueven desde la fuente que los genera a almacenes de datos con gran capacidad de espacio en disco. Este espacio en disco suele consumir recursos económicos, por tanto, se debe establecer una política asociada a la seguridad de la información Chuvakin y Schmidt (2012), del tiempo que deben permanecer en custodia y luego ser eliminados, en tabla II se muestra algunos aspectos importantes a considerar:

Tabla II. **Conservación de archivos de registro**

Requerimiento	Descripción del requerimiento
Estándares	Existen estándares internaciones como PCI con directrices según el tipo de institución.
Administración del riesgo	Según la data que se recolecte y el tipo de datos, el área de riesgos debe definir cuándo puede eliminarse un archivo.
Espacio	Se debe hacer una proyección del espacio que ocuparan los archivos a futuro, y si se cuenta con ese espacio.
Tipo de almacenamiento	Se debe considerar si la organización tiene la capacidad de costear las unidades de almacenamiento físico y si estas satisfacen las necesidades del sistema de gestión de archivos de *logs*. Puede tener en cuenta una opción de almacenamiento en la nube, con costos de mantenimiento menor a una instalación propia.

Fuente: elaboración propia.

7.1.3. Exploración de data en los archivos de registro

El análisis de eventos requiere que la data esté concentrada en un solo lugar para una exploración unificada, que pueda convertir data fragmentada en información de valor para la organización. Para este proceso se usan técnicas de análisis de información Kent y Souppaya (2006), descrita en la tabla III, sobre toda la data recolectada de los archivos generados en sus fuentes primarias.

Tabla III. **Técnicas de análisis de información**

Técnica	Descripción
Selección por tipo	Se establece un criterio bajo el cual se agrupan ciertos tipos de archivos de eventos.
Estadística	Se aplica estadística descriptiva sobre toda la data con el objetivo de encontrar parámetros para analizar su comportamiento.
Correlación de *logs*	Se establece un criterio para relacionar eventos, de distintas fuentes y tipos, pero que están relacionados.
Minería de datos	Se crea un modelo para búsqueda de patrones de eventos en la data recolectada.

Fuente: elaboración propia.

La detección de fallas es solo un paso de la gestión de archivos de eventos, para generar valor se debe enlazar a un sistema de monitoreo y alertas, que muestre información útil del estado del sistema. Para esto se utilizará el componente *Kibana*, del ELK-Stack.

7.1.4. Fuentes

Cada uno de los sistemas que conforman un ecosistema basado en tecnología son considerados fuentes de generación de archivos de eventos Chuvakin y Schmidt (2012), para clasificar estas fuentes existen 2 criterios: tipo de mecanismo y tipo de sistema. Esta tarea está a cargo del componente *Logstash*, de *ELK Stack*.

7.1.4.1. Tipo de sistema

Según el tipo de sistema, una fuente de generación de log puede ser de 3 tipos Kent y Souppaya (2006), descritos en la tabla IV:

Tabla IV. **Clasificación de fuente por tipo de sistema**

Fuente generadora	Información generada
Dispositivos de seguridad y de red	Inicio y finalización de sesión.
	Datos de conexión a un servidor de la red
	Data de transferencia de información.
	Reinicio de sistema.
	Cambios de configuración.
Sistemas operativos	Autenticación.
	Inicio, apagado y reinicio del sistema.
	Inicio, apagado y reinicio de servicios.
	Falla de servicios
	Estatus de sistema
Aplicaciones	bitácora
	Bitácora de usuarios con alto nivel de acceso.
	Actividades críticas.
	Reconfiguraciones.

Fuente: elaboración propia.

7.1.4.2. Tipo de mecanismo

Chuvakin y Schmidt (2012) En la clasificación por mecanismo existen 2 categorías:

- *Push-based*-Envió de archivos al servidor central

En esta categoría están todas las fuentes que utilizan Eventos Windows, SNMP o Syslog.

- *Pull-based*-Recolección de archivos por el servidor central

Esta categoría reúne a todas las aplicaciones desarrolladas para ir directamente donde se genera el log, y recolectar la data para enviarla a una base de datos centralizada por medio de agentes.

7.2. *ELK Stack*

ELK Stack, es un grupo de productos de código abierto de *Elastic*, diseñado para tomar datos de cualquier tipo de fuente y formato y buscar, analizar y visualizar esos datos en tiempo real (Sachdeva, 2017). Implementar *ELK Stack* para la centralización de log de sistemas y aplicaciones daría una vista del funcionamiento de las aplicaciones en tiempo real, que se incluyan en este sistema.

Gil, *et. al.* (2016) Implementaciones previas de *ELK Stack* han demostrado que este *software* de administración de log permite identificar fallos de aplicaciones a través del procesamiento de log, además de ayudar en la

investigación de la resolución de estos por medio del análisis de grandes cantidades de información de distintas fuentes de origen.

7.2.1. Componentes del *ELK Stack*

Chhajed (2015) La plataforma ELK es una solución completa de gestión centralizada de *logs*, construida sobre una combinación de tres componentes de código abierto: *Elasticsearch*, *Logstash* y *Kibana*. El componente central de *ELK Stack* es *Elasticsearch*, un motor de análisis y búsqueda de código abierto distribuido. Está basado en Apache Lucene y está diseñado para escalabilidad horizontal. *Logstash* es un canal de recopilación, enriquecimiento y transporte de datos. La capacidad de integrar conectores con una infraestructura común le da a *Logstash* la capacidad de procesar múltiples tipos de registros, eventos y fuentes de datos no estructurados para su distribución en una variedad de resultados, incluido *Elasticsearch*. ELK se completa con *Kibana*, que es una plataforma de visualización de datos que permite la interacción con los datos a través de gráficos. *Kibana* puede dar una visión de los datos con paneles que aprovechan una variedad de visualizaciones disponibles. A continuación se hace una descripción más detallada de cada componente:

7.2.1.1. *Logstash*

Turnbull (2019) Es el motor central de flujo de datos de *ELK Stack* para recopilar, enriquecer y unificar todos sus datos sin importar el formato o el esquema. El procesamiento en tiempo real es especialmente poderoso cuando se combina con *Elasticsearch*, *Kibana* y Beats.

Logstash es esencialmente un marco integrado para la recopilación, centralización, análisis, almacenamiento y búsqueda. Es un *software* de código abierto que puede unificar dinámicamente datos de fuentes dispares y normalizar los datos según el destino, en este caso *Elasticsearch*. Permite transformar cualquier tipo de evento con una amplia matriz de complementos de entrada, filtro y salida, con muchos códecs nativos que simplifican aún más el proceso de ingestión de data. Existe una amplia gama de filtros que se pueden aplicados a los registros recopilados para transformar los eventos.

Logstash tiene una arquitectura extensible y existen varios complementos para los desarrolladores. Ofrece información casi en tiempo real inmediatamente en el índice o el tiempo de salida. Enviar estos registros de *Logstash* a *Elasticsearch* permite realizar una amplia gama de mapeos, agregaciones y búsquedas.

7.2.1.2. *Kibana*

Como lo describe Azarmi (2017) Es un componente del ELK-Stack para visualización y análisis diseñado para interactuar con la base de datos *Elasticsearch*. Realiza análisis de datos avanzados y visualización de información en varios tipos de gráficos, como tablas o mapas. Se puede utilizar para buscar, ver e interactuar data estructurada o no estructurada, almacenada en *Elasticsearch*.

Interpretar grandes volúmenes de datos es bastante intuitivo con *Kibana* mediante la sencilla interfaz para desarrollar y administrar tableros con información, que pueden mostrar cambios en tiempo real gracias a la alta velocidad de respuesta de *Elasticsearch*.

Según Prakash, Kakkar y Patel (2017) Existes algunos modelos de tableros que por la información que muestran sueles ser implementados para un LMS, con algunos cambios según el giro de negocios de la institución que lo utiliza.

- *Syslog evaluation*

Los *logs* generados por cada agente son enviados para ser *ingestados* y procesador por *Elasticsearch*. Este tablero debe mostrar cada uno de los *logs* recibidos exitosamente y la información relevante en ellos.

- *Log severity evaluation*

Según la información de cada log, este debe ser agrupado por una criticidad. Este tablero debe mostrar los grupos de log según su criticidad.

- *Home dashboard evaluation*

Muestra información general sobre aplicaciones monitoreadas. El panel de inicio consta de series de tiempo de evaluación, unidades de mayor impacto, indicador de gravedad, visualización de gravedad.

- *Application evaluation*

La evaluación de la aplicación explica los resultados y apariencia de la aplicación que se ha creado.

7.2.1.3. *Elasticsearch*

Dixit, Rogozin'ski, Kuc, y Chhajed (2017) es un motor de análisis, búsqueda y almacenamiento distribuido en tiempo real. Se puede usar para muchos propósitos, pero un contexto en el que sobresale es la indexación de flujos de datos semiestructurados, como registros almacenados en un log.

Proporciona un motor de búsqueda multihilo, con capacidad de búsqueda por texto completo y un API que utiliza documentos JSON. Se puede utilizar para búsquedas de texto completo, búsquedas estructuradas, analítica, o una combinación de los tres.

Una de sus características clave es la capacidad de buscar rápidamente indexando el texto que se buscará. Puede realizar búsquedas de texto completo, manejar sinónimos y calificar documentos por relevancia. Además, también puede generar análisis y agregación a partir de los mismos datos, en tiempo real.

8. PROPUESTA DE ÍNDICE DE CONTENIDOS

9. METODOLOGÍA

9.1. Tipo de estudio

Se realizará un estudio de tipo cuantitativo para responder a la pregunta principal: ¿cómo implementar un control de fallos preventivo para las aplicaciones que funcionan en canales electrónicos de una entidad bancaria?, para esto se desarrollará una investigación sobre generación y centralización de *logs*, almacenamientos de registros de estos log en base de datos y luego aplicar algoritmos de *machine learning* sobre la data recolectada que permitan identificar patrones para detectar fallas del sistema, en tiempo real.

9.2. Diseño

El diseño del estudio es experimental, porque se utilizará algoritmos de *machine learning* sobre la data recolectada de todos los *logs* del sistema para identificar patrones de fallos anteriores para ser correlacionados con los datos procesados en tiempo real para detectar fallas de forma preventiva.

9.3. Alcance

El alcance del estudio es descriptivo, porque se busca describir un sistema de monitoreo de sistemas para detectar fallos de forma preventiva para las aplicaciones que funcionan en canales electrónicos de una entidad bancaria y todos los componentes de este. Se pretende describir el proceso para poder identificar fallos de forma preventiva para minimizar el impacto en rendimiento del

sistema, beneficiando directamente a todos los usuarios de los sistemas de la entidad bancaria en la cual se implementará el sistema.

9.4. Variables

A continuación, en la tabla V, se describen las variables.

Tabla V. **Variables**

Variables	Definición	Subvariables	Indicadores
Fallas detectadas de forma preventiva	Alertas de fallo detectadas por el sistema de monitoreo de forma preventiva.	Rendimiento	• Utilización de CPU • Memoria RAM en uso • Memoria RAM Disponible • Capacidad del disco duro • Espacio disponible en Disco. • Velocidad de lectura en disco • Velocidad de escritura en disco • Velocidad de datos enviados • Velocidad de datos recibidos
		Aplicación monitoreada	• Cantidad de aplicaciones monitoreadas.
		Archivos procesados	• Cantidad de archivos procesados • Cantidad de log procesados por aplicación
		Almacenamiento	• Velocidad de respuesta de consultas • Latencia de las consultas • Tasa de solicitudes
		Disponibilidad	• Nivel de servicio
		Alertas	• Número de alerta de riesgo bajo • Número de alertas de riesgo medio • Número de alertas de riesgo alto.

Fuente: elaboración propia.

9.5. Fases del estudio

A continuación se describen y se detallan las fases de la investigación.

9.5.1. Revisión documental

Para entender las técnicas utilizadas para la detección de fallos es necesario, entender cuál es el comportamiento y origen de estas, para ello se realizará una revisión documental, para el diseño e implementación de un sistema de monitoreo de aplicaciones que permita identificar fallas en tiempo real, contempla la recopilación de información sobre los siguientes temas:

- Principios de generación de *logs.*
- Fuentes generadoras de log.
- Conceptos básicos para detección de fallas.
- Correlación de datos.
- Herramientas de *software* libre para analizar *logs.*
- Procesamiento y transformación de información.
- *Machine learning* para análisis avanzado de log con.

9.5.1.1. Selección de la herramienta

El proceso de selección de la herramienta se ajusta a los requerimientos y presupuesto de la institución bancaria en la cual se pretende implementar, evaluando los siguientes parámetros:

- Costo total de propiedad.
- Tipo de licencia y soporte ofrecido.
- Actualizaciones de la herramienta.

- Características principales de la solución.
- Documentación y capacitación de los usuarios.
- Extensibilidad de la solución.

9.5.2. Diseño del sistema de monitoreo de aplicaciones en tiempo real

En la fase de diseño se incluyen las siguientes tareas:

9.5.2.1. Definición de una política

El desarrollo de una política de retención de los *logs* en las fuentes y en la central de almacenamiento debe tomar en cuenta los siguientes parámetros:

- La existencia y requerimientos exigidos por estándares de cumplimiento adoptados por la institución bancaria donde se va a implementar la solución.

- El nivel de riesgo aceptado por la institución bancaria donde se va a implementar la solución con relación al periodo de tiempo en que se desee analizar la información contenida en los *logs*.

- Cantidad total de *logs* generados por las diversas fuentes, generalmente medida en bytes por unidad de tiempo.

- Mecanismos de almacenamiento y archivado de información disponibles en la institución bancaria donde se va a implementar la solución.

9.5.2.2. Diseño de la arquitectura

Las características de la institución bancaria donde se implementará la solución se ajustan a un diseño descentralizado. Se implementará la solución ELK mediante un recolector con *Logstash* que filtrará la información transmitida desde las fuentes de *logs* y almacenará los resultados en una instancia de *Elasticsearch*. Las fuentes se clasifican en dos tipos: fuentes que poseen una instalación local de *Logstash* (*Logstash*-forwarder) para la transmisión y fuentes que utilizarán un agente de transporte diferente de *Logstash* (*Logstash*-forwarder)

9.5.2.3. Construcción del modelo probabilístico

Construcción de un modelo probabilístico, que se alimentará de los datos almacenados para aprender el comportamiento normal, formando un patrón y cuando detecte valores que se desvían de este comportamiento, se informará como una anomalía. Se utilizará un sistema probabilístico frente a un sistema de reglas estáticas para reducir el número de falsos positivos.

9.5.3. Desarrollo de la solución

- Implementación de la arquitectura.
    - Instalación del servidor *Logstash.*
        - Configuración de *Logstash.*
    - Instalación del servidor *Elasticsearch*

        - Configuración de los índices de *Elasticsearch.*

    - Instalación del servidor de *Kibana.*

    - Configuración de las fuentes generadoras de *logs.*

        - Configuración de Internet Information Server.
        - Configuración de Microsoft SQL Server.
        - Configuración de Windows Server.
        - Configuración de la aplicación de gestión de tráfico ntopng.

- Implementación del modelo probabilístico.

- Creación y configuración del panel de control.

- Pruebas integrales de funcionamiento del sistema.

9.5.4. Experimentación

El objetivo que se persigue es la detección de anomalías en series temporales de información, esto se validará a través de experimentación en un ambiente controlado. El sistema va a ir aprendiendo cuál es su comportamiento normal, formando un patrón. Cuando existen valores que se desvían de este comportamiento, se informará como una anomalía. Esto se realiza construyendo un modelo probabilístico conforme el sistema va aprendiendo. La idea es utilizar un sistema probabilístico frente a un sistema de reglas estáticas para reducir el número de falsos positivos.

9.5.5. Análisis y monitoreo de eventos

El análisis y monitoreo de eventos constituye la etapa de consumo de la información obtenida mediante la gestión de las fuentes generadoras de *logs*. No obstante, existen múltiples herramientas que se pueden integrar con la solución elegida para evaluación, en el presente escenario se utilizarán:

- *Kibana*, al formar parte de la solución ELK para la exploración, visualización y creación de cuadros de mando.

- SEC (*Simple Event Correlator*), para la correlación de eventos.

9.6. Técnica de recolección de información

En siguientes incisos se describe la técnica de recolección de la información necesaria.

9.6.1. Recolección de *logs*

La recolección de *logs* es el paso previo para la gestión centralizada en el que los registros son importados de diferentes fuentes a través de la infraestructura informática y se recopilan en una única ubicación central. Mediante el uso de ciertos cargadores de ficheros o datos, se envían los registros desde un origen como aplicaciones, contenedores, bases de datos o cualquier otro sistema y los registros se agregan y almacenan en una ubicación central. Para la recolección de *logs* que alimentaran el sistema de monitoreo en tiempo real, se toman en consideración los siguientes aspectos:

9.6.1.1. Formato de *logs*

Se definirá un formato de log a cada aplicación, según su naturaleza basando en los más extendidos:

- Registros por línea con los datos separados por espacios, para *logs* que generan gran cantidad de registros en un archivo plano con el objetivo de optimizar tu tamaño en disco.

- JSON y XML, para aplicaciones con intercambio de datos.

9.6.1.2. Estándar

Según el entorno o el tipo de log que vaya a generarse, existen varios estándares que para facilitar la intercomunicación de elementos.

- *Syslog*

Se utilizará para la gestión del sistema y la auditoría de seguridad, mensajes generales de información, análisis y depuración. Permite la separación del *software* que genera mensajes, el sistema que los almacena y el *software* que los informa y analiza.

- *Common Log Format* (CLF) *y Extended Log Format* (ELF)

Son formatos de archivo de texto estandarizado utilizado por servidores web para generar registros, la diferencia entre ambos es que ELF puede contener más información.

9.6.1.3. Definición de la información consignada en *logs*

Los datos mínimos que se deben guardar para analizar y obtener información válida son los siguientes:

- Fecha y hora, definir zona horaria.
- Tipo: error o alerta.
- Dirección IP del dispositivo origen.
- Código único del suceso.
- Mensaje con la descripción ampliada del suceso.
- Usuario del sistema o la aplicación.
- Contexto (fichero, página, parte de la aplicación que ha generado el registro o data ingresada en la petición).

9.6.1.4. Centralización de datos

Se realizará un proceso de centralización de *logs*, en el cual los registros son importados de diferentes fuentes a través de la infraestructura ELK-Stack y se recopilan en una única ubicación central, en este caso *Elasticsearch*. Mediante el uso de ciertos *Logstash*, se envían los registros desde un origen como aplicaciones, contenedores, bases de datos o cualquier otro sistema y los registros se agregan y almacenan en *Elasticsearch*.

9.6.1.5. Log indexer

Es una técnica de procesamiento de texto para analizar los registros en cada y transformarlos en datos que puedan ser almacenados e indexados para su posterior análisis.

10. TÉCNICAS DE ANÁLISIS DE INFORMACIÓN

10.1. Correlación de datos

La correlación es uno de los elementos importantes en la conformación de un analizador de *logs* el cual incluye también el proceso heurístico que este trae implícito pues se podría dar que un sensor no pueda tener la capacidad de detectar y recopilar información mientras que otro sensor si, por lo que es necesario que esta acción permita vincular estados en distintos sitios que conforman el sistema pero concatenados en un solo análisis que resuma el estado general y muestre las fortalezas o debilidades en números y gráficas para que al detectar una anomalía en el sistema se pueda saber cuál es el alcance negativo que pueda tener esta anomalía la institución bancaria y cuantas fuentes de datos, dispositivos o aplicaciones podrían verse afectados.

La validez de la correlación de datos toma mayor precio cuando existen fallos nunca vistos en los cuales es probable que no se tenga ningún tipo de información relacionada y se deba estar preparado para tener las medidas y herramientas suficientes para contrarrestar este tipo de fallas.

10.2. Análisis de *logs* por medio de *machine learning*

Para el análisis de *logs*, los agentes encargados del monitoreo usualmente pueden analizar los registros de aplicaciones para identificar actividad inusual, estos registros pueden contener información sobre eventos del sistema, acciones realizadas por los componentes del sistema operativo de diversos dispositivos (por ejemplo, apagar el sistema, iniciar un servicio, entre otros) registros de

auditoría, que contienen información de eventos de seguridad como intentos de autenticación fallidos, exitosos y cambios en la política de seguridad, eventos de aplicación, que son acciones operacionales significativas realizadas por aplicaciones, como inicio y cierre de aplicaciones, fallas de aplicaciones y cambios de configuración en aplicaciones principales.

Para el análisis de *logs* se usará un modelo a partir de la información suministrada para poder generar conclusiones automáticamente basado en el comportamiento de los datos a través de tendencias y periodicidad en tiempo real para identificar problemas más rápidamente.

10.3. Categorización de alertas

Se desarrollará un sistema de notificación basado en reglas sobre los eventos categorizados según su criticidad. Cada tipo de alerta muestra cuando se determina la posibilidad de ocurrencia una falla, por el modelo probabilístico, basado en el análisis de la data almacenada que incluye los eventos recopilados en tiempo real.

- Amarilla

Alerta con impacto bajo. Su mantenimiento puede ser programado o si está dentro de los parámetros aceptables para administrador, puede omitirse.

- Naranja

Alerta con impacto medio. Su mantenimiento puede ser programado, pero si no atiende puede escalar en su criticidad.

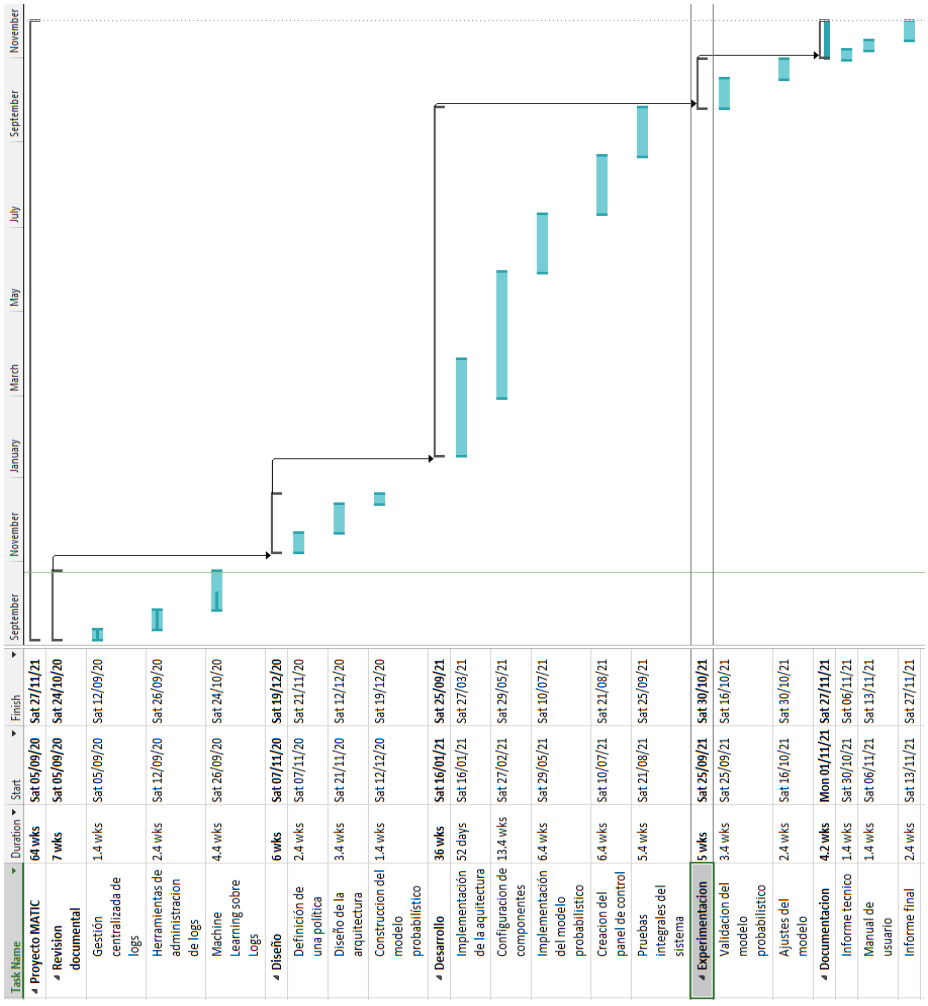
- Roja

Alerta con impacto crítico. Se debe atender inmediatamente para evitar que el sistema se vuelva inestable, por este motivo debe enviar un correo electrónico a la lista de administradores del sistema con toda la información relacionada al evento.

11. CRONOGRAMA

En la figura 5, se muestra el detalle del cronograma para el desarrollo del proyecto, el cual tendrá una duración de 64 semanas.

Figura 5. **Cronograma de actividades**



Fuente: elaboración propia.

12. FACTIBILIDAD DEL ESTUDIO

La idea del proyecto es implementar un sistema de monitoreo de aplicaciones que funcionan en canales electrónicos para una entidad bancaria que permita detectar fallos de forma preventiva. El desarrollo e implementación es posible debido a que la institución bancaria cuenta con la arquitectura y personal operativo en el área de mantenimiento de aplicaciones necesarios para el éxito del proyecto. Respecto a los costos de desarrollo, serán desarrollados por mi persona, sin que esto sea cargado a la institución financiera como una colaboración para desarrollo del proyecto de tesis MATIC.

12.1. Factibilidad operativa

La factibilidad operativa es la herramienta que demuestra que la investigación si es viable.

12.1.1. Recursos técnicos

El área de implementación y monitoreo de soluciones cuenta con personal técnico con el conocimiento sobre de tarea de mantenimiento de *software* y tienen la experiencia del funcionamiento las aplicaciones. Este personal está en la disposición de colaborar con información para el diagnóstico de fallos en los sistemas y las soluciones que actualmente aplican, para enriquecer el modelo probabilísimo de detección de fallos preventivo.

12.1.2. Recursos organizacionales

La institución bancaria cuenta con el personal con el conocimiento técnico y operativo para realizar el análisis para el desarrollo del sistema de monitoreo para aplicaciones que funcionan en canales electrónicos que permitan crear un modelo probabilístico que detecte efectivamente las fallas del sistema de forma preventiva.

Este personal está en la disposición colaborar con el tiempo para el desarrollo y capacitación, posterior al desarrollo del proyecto.

12.1.3. Acceso a la información

Se garantiza que el *software* tendrá acceso a las fuentes generadoras de *logs*, tanto transaccionales como históricas, para alimentar su base de datos de eventos antes de la implementación de la solución, y a partir de inicio de operaciones de esta.

12.1.4. Seguridad de la información

La información consigna en los *logs* recolectados por el sistema de monitoreo están sujetas a las políticas de seguridad de la institución bancaria y su uso fuera de esta está totalmente prohibida sin el permiso explícito de la institución.

El estudio es operativamente factible porque institución bancaria está en la disposición de asignar al personal administrativo y técnico durante y posterior a la implementación del sistema, asignar los permisos según las políticas de acceso y seguridad de la información para garantizar la confidencialidad de los

datos procesados y proporcionar todo el apoyo necesario para la realización del estudio.

12.2. Factibilidad técnica

La factibilidad técnica está enfocado en el recurso humano.

12.2.1. Personal operativo

La institución bancaria cuenta con una dirección encargada del mantenimiento del *software* después del despliegue cada solución, y dentro de la misma existe una coordinación que dedica exclusivamente al monitoreo en ambiente productos de sistemas y aplicaciones. Esta área será la encargada de administrar la solución de *software* que pretende ayudar en sus tareas de optimización de rendimiento y corrección de fallos, y cuentan con todo el conocimiento técnico y operativo de corrección de errores, mejoras de las capacidades, eliminación de funciones obsoletas y optimización.

12.2.2. Recursos tecnológicos

La institución bancaria cuenta con la infraestructura necesaria para la implementación del sistema de monitoreo de aplicaciones que funcionan en canales electrónicos para la detección de fallas de forma preventiva. Tiene la capacidad de crear un servidor virtualizado con las características técnicas solicitadas, red interna, firewall, *software* y certificados de seguridad para garantizar el correcto y óptimo funcionamiento. También cuenta con el personal operativo capacitado para operar y administrar dicho sistema.

12.2.3. Infraestructura

Debido a las políticas de acceso a la información y confidencialidad de los datos, la solución se instalará en un servidor virtual, dentro de la red de institución bancaria para tener acceso a todas las fuentes generadoras de *logs* que funcionan dentro de la red.

12.2.4. Equipo

La instalación de ELK – Stack necesitará lo siguiente:

- Un servidor de Ubuntu 20.04, con 4 GB de RAM, 500 GB de almacenamiento y 2 CPU configuradas con un usuario con privilegios de administrador. Este servidor estará virtualizado para que sus capacidades puedan ser escaladas a demanda, según lo requiera el sistema de monitoreo.

- OpenJDK 11 instalado.

- Nginx para proxy inverso para *Kibana*.

- Certificado TLS o SSL, para garantizar la seguridad del servidor.

El estudio es técnicamente factible porque la institución bancaria cuenta con una gerencia de tecnología, dedicada a dar soporte a todas las operaciones de su red de agencias y canales electrónicos, con la capacidad de crear, configurar y administrar los servidores donde funcionara el sistema y otorgar los permisos a través de la red interna que sean necesario. También cuenta con el personal

técnico capacitado para la administración y mantenimiento, posterior a la implementación de la solución.

12.3. Factibilidad económica

Está enfocada en demostrar que el estudio no requiere de muchos recursos financieros.

12.3.1. Modelo de suscripción

Elastic Stack impulsa una variedad de casos de uso, con distintos planes flexibles según las necesidades de cada empresa enfocados en maximizar las suscripciones en las estalaciones. Maneja una filosofía basada en recursos, donde se paga por lo que se usa a cualquier escala, como se describe en la tabla VI:

Tabla VI. **Suscripciones del *Elastic Stack***

Plan	Descripción	Características
Open Source	Apache 2.0: Ahora y siempre.	• Agrupación y alta disponibilidad • Potente búsqueda y análisis • Visualización y dashboards de datos
Basic	El plan gratis para siempre.	• Características de seguridad fundamentales del *Elastic Stack* • Capacidades como Elastic APM, Security, App Search, Workplace Search y Maps • Canvas y Lens • Alertas y acciones en el stack de *Kibana*3
Oro	Más características. Soporte dedicado.	• Reporting • Acciones de alertas de terceros de *Kibana*3 • Watcher • Gestión de ingesta • Soporte en horario comercial

Continuación tabla VI.

Platino	Funcionalidad avanzada. Soporte a toda hora.	• Características de seguridad avanzadas del *Elastic Stack* • Replicación entre clusters • Soporte 24/7/365
Enterprise	Orquestación del stack y protección de endpoint predeterminados.	• Acceso a Elastic Endgame2 • Acceso a características de orquestación de ECE y ECK

Fuente: elaboración propia.

12.3.2. Costo de desarrollo

El desarrollo del estudio requiere de un equipo multidisciplinario, con los siguientes perfiles:

- Analista de sistemas
- Desarrollador de *software*
- Gestor de proyectos
- Analista de QA
- Administrador de sistemas

De acuerdo con el cronograma, se realizó una estimación de tiempo en horas necesario para la finalización de cada entregable y se estimó una proyección basada en los precios a octubre 2020 en mercado de desarrollo de software en Guatemala del costo por hora según cada actividad, descrita en la tabla VI.

Tabla VII. **Costo de desarrollo**

Entregable	**Horas necesarias**	**Costo por hora**	**Costo total**
Revisión documental	328	$ 11.00	$ 3,608.00
Diseño	288	$ 11.00	$ 3,168.00
Implementación	1472	$ 18.00	$ 26,000.00
Experimentación	232	$ 8.00	$ 1,856.00
Documentación	208	$ 2.00	$ 416.00
Total			$ 35,048.00

Fuente: elaboración propia.

El estudio es económicamente factible porque se seleccionará el modelo de suscripción Basic, sin costo de licenciamiento por uso del *software* y para la arquitectura, la institución bancaria cuenta con una infraestructura capaz de mantener un servidor con las características solicitadas en su red interna que incluye el ecosistema de seguridad que actualmente posee. Para costo del personal, la institución está de acuerdo en ceder horas laborales dedicadas a la asesoría, configuración, implementación y administración del sistema de monitoreo para alcanzar los objetivos propuestos en el estudio.

Dado que se cubren los aspectos técnicos, operativos y económicos, necesarios el desarrollo del proyecto, se concluye que la realización del estudio es factible y se procederá según las fechas detalladas en el cronograma de actividades.

13. REFERENCIAS

1. Almeida, E., Koga, I., Santana, M., Guimaraes, P., Sugawara, L., y Eklin, T. (3 de agosto de 2017). Exploratory study of the *ELK Stack* for meteorological observation system data analysis. *Journal of Computational Interdisciplinary Sciences 8*(3), 131-142. Recuperado de https://www.researchgate.net/publication/323975183_Exploratory_study_of_the_ELK_stack_for_meteorological_observation_system_data_analysis.

2. Azarmi, B. (2017). *Learning Kibana 5.0*. Birmingham, UK: Packt Publishing.

3. Chhajed, S. (2015). *Learning ELK Stack*. Birmingham, UK: Packt Publishing.

4. Chuvakin, A. y Schmidt, K. (2012). *Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management*. Estados Unidos: Syngress.

5. Dixit, B., Rogozin'ski, M., Kuc, R. y Chhajed, S. (2017). *Elasticsearch*: A Complete Guide. Packt Publishing. Birmingham, UK: Packt Publishing.

6. Gil, J., Reveco, J. y Shen, T. (26 de julio de 2016). Operational *logs* analysis at ALMA observatory based on *ELK Stack*. *Proceedings of the SPIE, 9913*(23), 1-9. Recuperado de https://www.spiedigitallibrary.org/conference-proceedings-of-spie/9913/991323/Operational-*logs*-analysis-at-ALMA-observatory-based-on-ELK-stack/10.1117/12.2232258.full.

7. Kent, K. y Souppaya, M. (3 de septiembre de 2006). Guide to Computer Security Log Management. *NIST*, *800*(92), 1-72. Recuperado de https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf.

8. Mitra, M. y Sy, D. (17 de novienmbre de 2016). The Rise of *Elastic Stack*. DOI:10.13140/RG.2.2.17596.03203. Recuperado de https://www.researchgate.net/publication/309732494_The_Rise_of_Elastic_Stack.

9. Novak, J., Krajnc, A. y Zontar, R. (15 de julio de 2010). Taxonomy of Static Code Analysis Tools. *The 33rd International Convention MIPRO, 1*(1), 418-422. Recuperado de https://www.semanticscholar.org/paper/Taxonomy-of-static-code-analysis-tools-Novak-Krajnc/13a690909731c81a43d736e54d02994430fdb2ac.

10. Prakash, T., Kakkar, M. y Patel, K. (2 de octubre de 2017). Geo-Identification of Web Users through *Logs* using. *6th International Conference - Cloud System and Big Data Engineering,* 606-610. Doi: 10.1109/CONFLUENCE.2016.7508191. Recuperado de https://www.researchgate.net/publication/305675550_Geo-identification_of_web_users_through_*logs*_using_ELK_stack/citation/download.

11. Rochim, A., Aziz, M. y Fauzi, A. (12 de octubre de 2019). Design Log Management System of Computer Network Devices. *ICECOS*, 338-342. Doi: 10.1109/ICECOS47637.2019.8984494. Recuperado de https://www.researchgate.net/publication/339095357_Design_Log_Management_System_of_Computer_Network_Devices_Infrastructures_Based_on_ELK_Stack.

12. Sachdeva, G. (2017). *Practical ELK Stack: Build Actionable Insights and Business Metrics Using the Combined Power of Elasticsearch, Logstash, and Kibana*. New Delhi, Delhi, India: Apress.

13. Turnbull, J. (2019). *The Logstash Book, Log management made easy*. Maharashtra, India: Shroff Publishers.

14. West-Brown, M., Stikvoort, D., Kossakowski, K., Killcrece, G., Ruefle, R., y Zajicek, M. (1998). *Handbook for Computer Security Incident Response Teams (CSIRTs)*. Pittsburgh, PA, USA: Carnegie Mellon. Recuperado de https://resources.sei.cmu.edu/asset_files/Handbook/2003_002_001_14102.pdf

15. Yang, C., Kristiani, E., Wang, Y. y Liu, M. (2019). *The Implementation of NetFlow Log System Using Ceph and ELK Stack*. New York, USA: Springer Link.