OFICINA DE POSGRADOS

Tema:

ESTÁNDARES DE SEGURIDAD EN EL CICLO DE VIDA DE APLICACIONES USANDO CONTENEDORES

Proyecto de investigación previo a la obtención del título de Magister en Ciberseguridad

Línea de Investigación:

SEGURIDAD DE LA INFORMACIÓN

Autor:

ING. FREDY LEONARDO ARROBA FLORES

Director:

ING. MG. GALO MAURICIO LÓPEZ SEVILLA

Ambato – Ecuador

Septiembre 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

HOJA DE APROBACIÓN

Tema:

ESTÁNDARES DE SEGURIDAD EN EL CICLO DE VIDA DE APLICACIONES USANDO CONTENEDORES

Línea de Investigación:

SEGURIDAD DE LA INFORMACIÓN

Autor:

ING. FREDY LEONARDO ARROBA FLORES

Galo Mauricio López Sevilla, MSc. f.
CALIFICADOR

José Marcelo Balseca Manzano, MSc. f.
CALIFICADOR

Paúl Hernán Zurita Llerena, MSc. f.
CALIFICADOR

Juan Carlos Acosta, Padre, MSc. f.
DIRECTOR UNIDAD ACADÉMICA

Hugo Rogelio Altamirano Villarroel, Dr. f.
SECRETARIO GENERAL PUCESA

Ambato – Ecuador

Septiembre 2021

DECLARACIÓN DE AUTENCIDAD Y RESPONSABILIDAD

Yo: **FREDY LEONARDO ARROBA FLORES**, con CC. **180449565-1** autor del trabajo de graduación intitulado: "ESTÁNDARES DE SEGURIDAD EN EL CICLO DE VIDA DE APLICACIONES USANDO CONTENEDORES", previa a la obtención del título profesional de **MAGISTER EN CIBERSRGURIDAD**, en la **OFICINA DE POSGRADOS.**

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad.

Ambato, septiembre 2021

…….

FREDY LEONARDO ARROBA FLORES

CC. 180449565-1

AGRADECIMIENTO

A Dios por darme la salud y la fuerza para cumplir mis objetivos de vida, a mis hermanos y mi tía por estar siempre pendientes y apoyarme en todo el proceso de la maestría en Ciberseguridad y a todos quienes conforman la Maestría en Ciberseguridad de la Pontificia Universidad Católica del Ecuador Sede Ambato – Primera Cohorte.

A mi tutor Ing. Mg. Galo Mauricio López Sevilla, quien me guio en el desarrollo del trabajo de titulación para cumplir mi objetivo planteado. Mi hijo y padres que son mi motor en la vida, siempre están en todo momento con sus palabras de aliento, para darme fuerzas y cumplir mis objetivos. También, a MSc. Teresita Freire coordinadora de la maestría, quien siempre estuvo pendiente en todo el proceso de la maestría en Ciberseguridad.

DEDICATORIA

A Dios por haberme permitido cumplir un objetivo más en mi vida profesional y darme las fuerzas y conocimiento ante obstáculos que se han presentado día tras día, con la salud y sabiduría para lograr mis objetivos.

A mis padres Gilbert y Corina por el apoyo incondicional en toda mi vida estudiantil y profesional, a mis hermanos Darwin y Mariela por estar siempre pendientes y apoyarme en todo momento.

A mi hijo amado Josue David, que es mi motor para cumplir los objetivos y llegar a ser un padre ejemplar.

Leonardo

RESUMEN

En la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., se evidencia, que el monitoreo en el ciclo de vida de las aplicaciones se los realiza de forma manual; ocasiona pérdida de tiempo en los usuarios y la seguridad en las aplicaciones se deja en segundo plano, como la seguridad en la información, lo cual, empieza a perder confianza en los aplicativos existentes en la empresa.

La necesidad de aplicar estándares de seguridades en el ciclo de vida de las aplicaciones ya sea local o internacional, cada día es requerido por las empresas que manejan información digital. Con este antecedente, el objetivo del trabajo de investigación es implementar estándares de seguridad en el ciclo de vida de las aplicaciones con un ambiente escalable que utiliza contenedores. Para cumplir con el mismo, se hace uso de la metodología de investigación de tipo analítico-sintético que ayuda con la descomposición del objeto de estudio en cada una de sus partes para estudiarlas en forma individual y luego de forma integral como son las vulnerabilidades existentes en las aplicaciones. Para el desarrollo se hace uso de la metodología ágil *Kanban* que ayuda a controlar las tareas de la implementación de forma transparente y ordenada. Como resultado del presente trabajo, se espera tener estándares de seguridad en el ciclo de vida de las aplicaciones para mitigar vulnerabilidades y mantener la información segura en un ambiente escalable.

Palabras claves: Ciclo de vida, aplicaciones, mitigación, vulnerabilidades, contenedores, seguridad.

ABSTRACT

It is evident that at Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., monitoring of the life cycle of applications has been carried out manually; this causes users to lose time and the applications' safety is left (in the background) unattended, such as safety regarding information, which has led to a distrust in existing applicationsfrom the company.The necessity of applying security standards in the life cycle of applications, whether local or international,is required every day by companies that handle digital information. Based on this antecedent, the objective of this study is to implement security standards in the life cycle of applications with a scalable environment by using containers. To achive it, the analytical-synthetic research methodology is used to help the deconstruction/separation/division of the object of study in each of its parts to study them individually and then as a whole; (to see) this way, the existing vulnerabilities in the applications will be apparent. For the development, the Kanban methodology is used, which helps to control the implementation tasks in a transparent and orderly way. As a result of the research, it is expected to have safety standards in the application life cycle to mitigate vulnerabilities and keepinformation safe in a scalable environment.

Keywords: life cycle, applications, mitigation, vulnerabilities, containers, security.

ÍNDICE

PRELIMINARES

INTRODUCCIÓN

En la actualidad debido a la demanda de aplicaciones, las empresas ecuatorianas necesitan realizar desarrollos más rápidos y disponer de entregables en menor tiempo, por ello emplean metodologías ágiles en su ciclo de vida del desarrollo conocida en inglés como: *Systems Development Life Cycl*e (SDLC). Esto tiene innumerables ventajas, pero en muchas ocasiones se descuida la seguridad y calidad de los entregables, además, la cultura de las empresas hace que la seguridad sea un paso al final del ciclo de vida del desarrollo de software, limita las entregas y es cuestión de los expertos en seguridad. (Ricote, 2019)

En algunas industrias ecuatorianas, según Canalnews (2020) se ha evidenciado una falla de seguridad que permite a los ciberdelincuentes manipular datos, cambiar la configuración de privacidad del usuario y extraer datos personales almacenados en estas cuentas. El desarrollo de las aplicaciones actualmente está enfocado a la funcionalidad y la seguridad no se considera dentro de la estructura del desarrollo, esto provoca brechas de seguridad en las aplicaciones ya sea en ambientes de pruebas o de producción.

Actualmente existe una variedad de herramientas para el monitoreo del rendimiento de las aplicaciones. Manger (2020) menciona que se emplean casi siempre para monitorear el rendimiento de las aplicaciones en entornos productivos, un monitoreo de aplicaciones en el entorno de desarrollo disminuirá significativamente el tiempo de desarrollo y aumentar el rendimiento y estabilidad de las aplicaciones pero no se enfocan en el monitoreo para la seguridad en el ciclo de vida de las aplicaciones, por lo se lo realiza de forma manual o en el peor de los casos no se lo realiza y solo se enfocan a la funcionalidad.

A nivel de seguridad, se considera a los productos software como entes vivos en constante cambio para corregir vulnerabilidades, añadir controles y adaptarse a las regulaciones y amenazas cambiantes.

Las aplicaciones necesitan ser publicadas dentro de una infraestructura, según Magic Online (2021) las ventajas inmediatas de alquilar un servidor en la nube, y con la posible ayuda del host, aprenderán a compartimentar el espacio del servidor en la nube para privatizar el espacio dedicado para la empresa.

> Los Contenedores son métodos centrados en las aplicaciones para ofrecer aplicaciones escalables de alto rendimiento, en cualquier infraestructura que se elija. Los Contenedores son los más adecuados para ofrecer microservicios, al proporcionar entornos virtuales portátiles y aislados para que las aplicaciones se ejecuten sin interferencia de otras aplicaciones en ejecución. (Serrotho, 2020)

En el contexto internacional, se encuentra el trabajo de Conislla (2020), en donde indica que los negocios digitales son capaces de generar millones de dólares de ganancia mediante el uso estratégico del software. Sin embargo, este gran rédito se ve opacado por los millones que pierden en robos y daño reputacional causado por los ataques de cibercriminales. Por ello, la seguridad en el desarrollo y operación del software constituye un factor clave para garantizar la rentabilidad de los negocios.

Por otra parte, en el contexto nacional, Guerra Terán (2020)**,** menciona que el mercado de desarrollo de aplicaciones en Ecuador está en expansión. Hay empresas dedicadas al desarrollo de aplicaciones móviles y emprendedores que hacen aplicativos de sus ideas, agrega que las aplicaciones internacionales tienen mayor inversión en marketing y publicidad, esa es su principal ventaja frente a las opciones locales.

> Es así como, en el Ecuador existen pocas empresas que aplican seguridad en el ciclo de vida del desarrollo de software y la seguridad no es considerada como un requerimiento del desarrollo, debilidades que son aprovechadas por los hackers. Por tanto, el uso inadecuado de las herramientas de monitoreo del ciclo de vida de las aplicaciones en empresas hace que tengan problemas de seguridad para la protección de privacidad digital que se aplican para evitar el acceso no autorizado a los datos y al no tener medidas de seguridad los costos para solucionar cualquier problema son mayores cuanto más tarde se detecte. (PowerData, 2020)

Según WMWARE (2020), las aplicaciones modernas son vulnerables a los hackers y programas maliciosos. Los ciberataques son más fáciles que nunca y las pérdidas por problemas de seguridad está en aumento. En la actualidad, uno de los temas de mayor importancia y trascendencia de toda la sociedad, es la seguridad y resguardo de la información, más aún en empresas que realizan transacciones en línea.

> El sistema de gestión de seguridad de la información persigue la protección de la información y de los sistemas que la manejan, del acceso con la finalidad de proteger la confidencialidad, la integridad y la disponibilidad de la información sensible de la organización. Los ambientes donde se despliegan las herramientas para el monitoreo de vulnerabilidades de las aplicaciones no son escalables, hoy en día se utiliza la tecnología de contenedores, de forma que, ayuda a que las aplicaciones sean escalables en cualquier ambiente de desarrollo. (SGSI, 2015)

> En la mayor parte de empresas no consideran a la seguridad de las aplicaciones como de vital importancia dentro de la organización más bien le toman como una necesidad en segundo plano, por este motivo existe un mayor porcentaje de ataques en las vulnerabilidades de las aplicaciones. (Ataques Web, 2018)

Con la llegada del COVID-19, muchas empresas tuvieron que recurrir al uso intensivo de herramientas digitales para implementar el teletrabajo, realizar compras y ventas online, así como gestionar procesos de producción de forma remota la mayor parte de empresas digitalizaron sus negocios por medio de plataformas en línea sin mayor seguridad en sus aplicaciones, da lugar a los ataques de ciberdelincuentes. (Henriquez, 2020)

Con base en lo expuesto, la pregunta central que rige esta investigación es: ¿Cómo mejorar la seguridad dentro del ciclo de vida de las aplicaciones en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda.?, la cual, se derivan las siguientes preguntas complementarias:

¿Qué aportes teóricos se han realizado a nivel local en el desarrollo de aplicaciones seguras?

¿Cómo se desarrolla el proceso del ciclo de vida para aplicaciones que implementan estándares de seguridad?

¿Cuál es la incidencia de implementar estándares de seguridad en el ciclo de vida de las aplicaciones por medio de contenedores?

Para responder a las preguntas descritas, se plantea como objetivo de este trabajo el implementar estándares de seguridad en el ciclo de vida de aplicaciones que usa contenedores. La consecución del objetivo se logrará gracias al cumplimiento de los siguientes objetivos específicos:

1. Fundamentar usos de estándares de seguridad para el aseguramiento del ciclo de vida de las aplicaciones.
2. Aplicar estándares de seguridad en el ciclo de vida de aplicaciones ejecutado por contenedores.
3. La implementación de estándares seguridad por medio de contenedores, evidencia la seguridad en el proceso del ciclo de vida de aplicaciones.

La metodología ágil que se va a aplicar para lograr este proyecto es *Kanban*, su objetivo principal es gestionar de manera general como se completan tareas, se basan en el desarrollo incremental, para dividir el trabajo en partes para una mejor gestión de proyectos. Se espera como resultado tener un sistema centralizado para el control de notificaciones con gráficos estadísticos que ayudará a mitigar ataques en el ciclo de vida de aplicaciones en un ambiente escalable.

Según ISW (2015), Kanban es una metodología *Just In Time* (JIT), sus principales ventajas son:

- Reducción del trabajo en progreso.
- Control del flujo de trabajo (fácil localización de cuellos de botella, y control sencillo del estado de desarrollo del producto).
- Fácil de aplicar (apenas existen reglas y se llevara a cabo de forma artesanal, no requiere software adicional).
- Promueve el trabajo en equipo (los desarrolladores se comunican y se ayudan si alguna actividad da problemas).
- Reducción del tiempo perdido (todo el desarrollo se hace bajo demanda, por lo que no hay partes que sobren, además, si alguien acaba su trabajo ayuda al resto del equipo).
- Facilidad para añadir mejoras (el tablero permite que todo el equipo de desarrollo vera cómo se realiza el producto desde una visión global, lo que permitirá reflejar a sus compañeros diversas mejoras sobre el mismo).

Para llevar a cabo la investigación, es necesario definir la metodología de investigación a seguir, la misma que se considera como una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que

> se seguirán durante el desarrollo de un proceso de investigación para la producción de conocimiento, es decir, son los pasos que guían un proceso con el fin de llevar una ejecución ordenada. (Coelho, 2019)
>
> Al respecto, el método de investigación general a utilizar es el método analítico-sintético es aquel método de investigación que consiste en la desmembración de un todo, descomponiéndolo en sus partes o elementos para observar las causas, la naturaleza y los efectos y después relacionar cada reacción mediante la elaboración de una síntesis general del fenómeno estudiado. (Sosa, 2013)

El método analítico ayuda a examinar, descomponer o estudiar minuciosamente una cosa. Por tanto, el método analítico comienza con el todo de un fenómeno y lo revista parte por parte, mientras que el método sintético es la composición de un todo mediante la unión de sus partes, el método sintético, por lo tanto, es aquel que procede de lo simple a lo complejo, donde las partes simples que se separaron en el análisis, una vez revisadas, ahora son integradas por la síntesis.

Actualmente en las empresas ecuatorianas, para poder determinar la existencia de un ataque o vulnerabilidad de seguridad informática en las aplicaciones, no se cuenta con un sistema automático; sino, la identificación de un problema de seguridad se realiza de forma empírica. Es importante estudiar la seguridad en las aplicaciones que se encuentran dentro de un ciclo de vida, tanto en desarrollo como en producción, puesto que este estudio de manera global permite mantener una visión general de las posibles vulnerabilidades y sus soluciones de forma oportuna y segura. En este contexto la tecnología de contenedores juega un papel importante, pues facilita escalar a medida, genera desarrollos y ajustar a las necesidades en función del tiempo durante el ciclo de vida de las aplicaciones. Se espera como principal resultado tener un sistema centralizado para el control de notificaciones con gráficos estadísticos que ayudará a mitigar ataques en el ciclo de vida de aplicaciones en un ambiente escalable. En la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., se tendrá un monitoreo constante de las aplicaciones que se encuentra en producción y en ambiente de pruebas, las alertas de una vulnerabilidad se presentarán de forma gráfica, para tomas de decisiones efectivas.

El proyecto aportará en el monitoreo de un ciclo de vida de las aplicaciones con herramientas libres, adicional se podría escalar con la integración continua, es una práctica de desarrollo de software, la cual, los programadores suben su código a un repositorio central donde automáticamente pasan las pruebas métricas y de calidad. Esta técnica se suele realizar regularmente para detectar fallos cuanto antes y así mantener el código siempre actualizado.

CAPÍTULO I. ESTADO DEL ARTE Y LA PRÁCTICA

Las aplicaciones han sido objeto de grandes cambios, lo cual, ha permitido proporcionar herramientas de última tecnología que presentan características sumamente innovadoras, según de su tipo, lenguaje, entorno de desarrollo y el sistema operativo en que vaya a ser ejecutado con la meta de facilitar el trabajo de sus usuarios, para ello, los desarrolladores se han visto obligados a utilizar estándares dentro del ciclo de vida de las aplicaciones para mejorar la calidad y tiempos de entrega.

El presente capítulo tiene como finalidad, proporcionar información fundamental para el proyecto sobre temas que se describen a continuación:

1.1 Ciclo de vida de aplicaciones

Con respecto al ciclo de vida de las aplicaciones, se podría interpretar los siguientes conceptos:

Tabla 1. Definición de Ciclo de vida de aplicaciones

Referencias	Conceptos
(Guévin, 2018, s.p.)	"El ciclo de vida de una aplicación se compone de todas las etapas que enfatizan su curso: desde la idea inicial hasta que los usuarios dejan de usarlo, y desde la concepción y la implementación de su actualización".
(Carrera, 2013)	Es la forma mediante, la cual, se describen los diferentes pasos que se seguirán para el desarrollo de un software, inicia desde una necesidad hasta llegar a la puesta en marcha de una solución y su apropiado mantenimiento. El ciclo de vida para un software comienza si se tiene la necesidad de resolver un problema, y termina el programa que se desarrolló para cumplir con los requerimientos deja de ser utilizado.
(Red Hat, 2020, s.p.)	"La gestión del ciclo de vida de las aplicaciones involucra a las personas, las herramientas y los procesos que gestionan el ciclo de vida de una aplicación desde que se diseña hasta que deja de estar disponible".

Fuente: elaboración propia

Se podría decir que el SDLC parte desde una necesidad o idea inicial hasta dejar de estar disponible, siempre va a estar en constante monitoreo desde la etapa de desarrollo hasta que la aplicación este en producción. El ciclo de vida cumple con varios pasos a seguir para obtener aplicaciones funcionales y operativas tal como se ejecutó el levantamiento de requerimientos.

Una aplicación nace, vive y muere; es por lo que se refiere a un SDLC, donde permite gestionar el ciclo de vida de una aplicación.

En el ciclo de vida de aplicaciones se tiene varias etapas como se muestra en el Grafico 1. Las cuales, ayudan a seguir pasos por cada etapa con el objetivo de entregar un producto funcional y de calidad en el menor tiempo posible.

Gráfico 1. Ciclo de vida de desarrollo seguro de software

Fuente: (Red Hat, 2020)

1.1.1 Paradigma de desarrollo de Software

El paradigma de desarrollo de software ayuda al desarrollador a escoger una estrategia para desarrollar el software. SIMPLYEASYLEARNING (2020) menciona que el paradigma de desarrollo software tiene su propio set de herramientas, métodos y procedimientos, los cuales, son expresados de forma clara, y define el ciclo de vida del desarrollo del software. Algunos paradigmas de desarrollo de software o modelos de proceso se definen a continuación:

Modelo de cascada

El modelo de cascada es el modelo de paradigma más simple en desarrollo de software. Sigue un modelo en que las fases del SDLC funcionarán una detrás de la otra de forma lineal. Lo que

significa que solamente si la primera fase se termina se podría empezar con la segunda, y así progresivamente.

Modelo repetitivo: Este modelo guía el proceso de desarrollo de software en repeticiones. Proyecta el proceso de desarrollo de forma cíclica se repite cada paso después de cada ciclo en el proceso de SDLC.

Modelo en espiral: El modelo en espiral es una combinación de ambos modelos, el repetitivo y uno del modelo SDLC. Se podría combinar un modelo de SDLC combinado con un proceso cíclico (modelo repetitivo).

Modelo V: El mayor inconveniente del modelo de cascada es que solo se pasa a la siguiente fase si se completa la anterior, por tanto, no es posible volver atrás si se encuentra algún error en las etapas posteriores. El Modelo V aporta opciones de evaluación del software en cada etapa de manera inversa.

Modelo *Big Bang*: Este modelo es el modelo con la forma más simple. Requiere poca planificación, mucha programación, también, muchos fondos. Este modelo se conceptualiza alrededor de la teoría de creación del universo *Big Bang.* Tal como cuentan los científicos, después del *big bang* muchas galaxias, planetas y estrellas evolucionaron. De la misma manera, si reunimos muchos fondos y programación, quizá podemos conseguir el mejor producto de software.

1.1.2 Herramientas para gestionar SDLC

Jira: Es un software que forma parte de una gama de productos diseñados para ayudar a equipos de todo tipo a gestionar el trabajo. En principio, según Atlassian (2019) Jira se diseñó como un gestor de incidencias y errores. Sin embargo, se ha convertido en una poderosa herramienta de gestión de trabajo para todo tipo de casos de uso, desde la gestión de requisitos y casos de prueba hasta el desarrollo de software ágil. En esta guía, se descubrirá las funciones y características de Jira capaces de ayudar a tu equipo con tus necesidades exclusivas.

Características

- Herramienta muy simple y a la vez muy potente.
- Permite una definición de filtros muy potente y versátil.

- Permite la definición de cuadros de mando (*dashboards*) con gran cantidad de widgets.
- Cada usuario podría definirse sus propios filtros y cuadros de mando, compartirlos o bien utilizar los definidos por otros usuarios.
- Ofrece la posibilidad de personalizar el aspecto de las pantallas (*look* & *feel*), esto permite adaptarlo a la imagen corporativa de la empresa.
- Posibilidad de definir diferentes tipos de incidencias/tareas. Éstas podrían vincular a un mismo flujo de trabajo (*workflow*) o a flujos de trabajo diferentes.
- Posibilidad de definir diferentes flujos de trabajo, en función de las necesidades.
- Definición de pantallas, campos personalizados y configuraciones de campos.
- Definición de estados de las incidencias.
- Definición de prioridades.

Elastic Stack: ELK (Elastic, Logstash y Kibana) o Elastic Stack, según García (2019) es un conjunto de herramientas *open source* desarrolladas por Elastic que permite recoger datos de cualquier tipo de fuente y en cualquier formato para realizar búsquedas, análisis y visualización de los datos en tiempo real.

Componentes

- **Elasticsearch:** motor de búsquedas distribuido construido sobre *Apache Lucene*. Permite indexar y recuperar documentos *JSON* (sin esquema rígido) en distintos formatos. Está basado en Java y proporciona un interfaz *RESTFUL* de acceso.
- **Logstash:** es un motor que permite recolectar datos de distintas fuentes, normalizarlos y distribuirlos. Originalmente optimizado para ficheros de logs, si bien podría leer datos de muchos tipos de fuentes.
- **Kibana:** herramienta que permite la visualización y exploración en tiempo real de grandes cantidades de datos. A través de la representación gráfica permite consultar de forma sencilla conjuntos complejos de datos.
- **Beats:** son cargadores de datos que se instalan en servidores y funcionan como agentes que envían información operacional a Elasticsearch.

1.2 Estándares de seguridad en aplicaciones

A continuación, se explica los estándares de seguridad en aplicaciones:

> Actualmente las seguridades en el desarrollo de aplicaciones no son tomadas en cuenta en el ciclo de vida, se la dejan en segundo plano, es importante aplicar seguridades en las aplicaciones, son principios de diseño y buenas prácticas a implantar en el SDLC, para detectar, prevenir y corregir los defectos de seguridad en el desarrollo y adquisición de aplicaciones, de forma que se obtenga software de confianza y robusto frente a taques maliciosos, que realice solo las funciones para las que fue diseñado, que esté libre de vulnerabilidades, ya sean intencionalmente diseñadas o accidentalmente insertadas durante su ciclo de vida y se asegure su integridad, disponibilidad y confidencialidad. (Figueroa, 2020)

Application Security Verification Standard (ASVS); o por su traducción al idioma español; Estándar para la Verificación de la Seguridad en Aplicaciones, es un esfuerzo comunitario para establecer un framework o marco de trabajo sobre requerimientos en seguridad y controles, los cuales, se enfocan en normalizar los controles de seguridad funcional y no funcional, requeridos si se diseña, desarrolla y prueban aplicaciones web modernas.

> Una de las mejores maneras de utilizar el estándar para la verificación de seguridad en aplicaciones, es utilizarlo para crear una lista de verificación para codificación seguridad específica para una aplicación, plataforma u organización. Adaptar ASVS a casos propios incrementará el enfoque sobre los requerimientos en seguridad, los cuales, son más importantes para los proyectos y entornos. (ReYDeS, 2018)

Niveles para la verificación de seguridad en aplicaciones

El estándar para la verificación de seguridad en aplicaciones define tres niveles de verificación para la seguridad, con cada nivel incrementa su profundidad.

- ASVS Nivel 1 es para todo el software.
- ASVS Nivel 2 es para las aplicaciones de datos sensibles, los cuales, requieren protección.
- ASVS Nivel 3 es para las aplicaciones más críticas, aplicaciones realizan transacciones de alto valor, tiene datos médicos sensibles, o cualquier aplicación el cual requiera un alto nivel de confianza.

Cada nivel del ASVS contiene una lista de requerimientos para la seguridad. Cada uno de estos requerimientos podrían, también, ser mapeado hacia características y capacidades específicas en seguridad, los cuales, construirán el software por los desarrolladores.

1.2.1 ISO 27034

El estándar ISO/IEC 27034 según CEC (2020), proporciona un estándar reconocido internacionalmente para la seguridad de las aplicaciones. Aunque todavía no se ha completado oficialmente, gran parte de la estructura del estándar ISO 27034 ya se establece a través de la publicación de la primera parte: ISO/IEC 27034–1: 2011.

El estándar proporciona orientación para ayudar a las organizaciones a integrar la seguridad en los procesos utilizados para gestionar sus aplicaciones y es aplicable a aplicaciones desarrolladas internamente, aplicaciones adquiridas de terceros y donde el desarrollo o la operación de la aplicación se subcontrata.

La ISO 27034 está estrechamente alineada con varias otras normas ISO, particularmente ISO 27005 para la gestión de riesgos de seguridad de la información.

Objetivo de ISO/IEC 27034

Esta norma multi-partes proporciona orientación sobre cómo especificar, diseñar/seleccionar y aplicar controles de seguridad de la información a través de un conjunto de procesos integrados a lo largo del SDLC de una organización. Está orientada a los procesos.

ISO/IEC 27034-1 se basa en los siguientes principios fundamentales:

- La seguridad es un requisito.
- La seguridad de las aplicaciones depende del contexto.
- Inversión apropiada para aplicaciones de seguridad.
- La seguridad en las aplicaciones estará demostrada.

El objetivo es asegurar que aplicaciones informáticas ofrezcan el nivel deseado o necesario de seguridad y ofrece orientación sobre seguridad de la información en el diseño, desarrollo, programación e implementación de sistemas de aplicación.

1.3 Objetivos de la Seguridad Informática

La tríada conocida como CIA es un modelo de seguridad de la información, muy utilizado, que podría guiar los esfuerzos y políticas de una organización para mantener sus datos seguros. El modelo no tiene nada que ver con la Agencia Central de Inteligencia de los Estados Unidos; más bien, las iniciales representan los tres principios en los que se basa la seguridad de la información:

Confidencialidad: Solo los usuarios y procesos autorizados podrán acceder o modificar los datos. Impide la divulgación de información a personas o sistemas no autorizados. A grandes rasgos, asegura el acceso a la información únicamente a aquellas personas que cuenten con la debida autorización.

Integridad: Los datos se mantendrán en un estado correcto y nadie modificara de manera incorrecta, ya sea accidental o maliciosamente. Busca mantener los datos libres de modificaciones no autorizadas. (No es igual a integridad referencial en bases de datos.) en general, la integridad es el mantener con exactitud la información tal cual fue generada, sin ser manipulada o alterada por personas o procesos no autorizados.

Disponibilidad: Los usuarios autorizados podrán acceder a los datos, es la característica, cualidad o condición de la información de encontrarse a disposición de quienes accedan a ella, ya sean personas, procesos o aplicaciones. A grandes rasgos la disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran.

Los tres principios son prioritarios para cualquier profesional de seguridad de la información. Pero considerarlos como una tríada obliga a que los profesionales de la seguridad deban realizar el difícil trabajo de pensar cómo es que se superponen y, a veces, cómo podrían oponerse entre sí, lo que ayudará a establecer prioridades en la implementación de las políticas de seguridad. (OnLine, 2020)

Gráfico 2. La triada de la seguridad

Fuente: (OnLine, 2020)

La información que se tiene en las aplicaciones estará con estándares de seguridad como se muestra en el Gráfico 2., con el objetivo de tener segura la información aplicar las tres reglas: confidencialidad, integridad y disponibilidad, en la actualidad la mayoría de las empresas ecuatorianas de desarrollo solo se enfocan a la funcionalidad de las aplicaciones y dejan en segundo plano la seguridad, es un factor importante e indispensable en el ciclo de vida de las aplicaciones.

1.4 Máquinas Virtuales y Contenedores

Las máquinas virtuales (VM) y los contenedores de Linux son entornos informáticos empaquetados que combinan varios elementos de TI y los aíslan del resto del sistema. Las principales diferencias radican en la capacidad de ampliación y la portabilidad.

1.4.1 Máquinas Virtuales

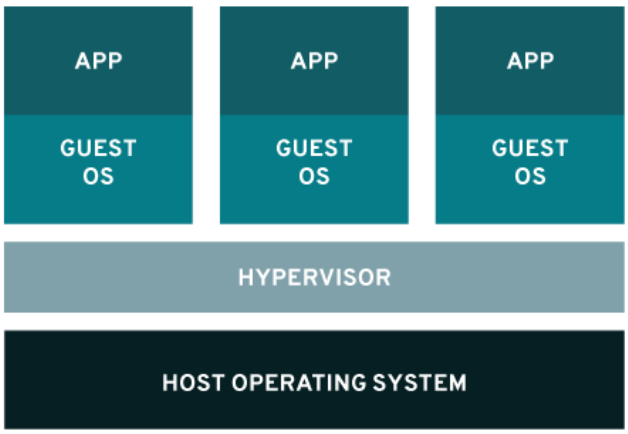
A continuación, se define que es una máquina virtual:

> Es un software que crea una capa independiente donde se emula el funcionamiento de un computador real con todos los componentes de hardware que necesita para funcionar (disco duro, memoria RAM, tarjetas de red, tarjeta gráfica, etc.) y que podrán ejecutar cualquier sistema operativo o programa, tal y como lo haría un ordenador real.

> Toda esta emulación se encapsula en una serie de archivos que actúan como contenedor desde el que se ejecuta la máquina virtual en una ventana del ordenador como si de un programa más se tratara y sin que nada de lo que suceda en el interior de esa ventana afecte al ordenador que la ejecuta. (Andrés, 2017)

La tecnología de VM permite compartir el hardware de modo que lo podrán utilizar varios sistemas operativos al mismo tiempo. Un esquema simplificado de su arquitectura se muestra en el Gráfico 3.

Gráfico 3. Esquema de una máquina virtual



Fuente: (Cuervo, 2019)

1.4.2 Contenedores

> Comparten el mismo *kernel* del sistema operativo, si bien tienen un espacio aislado y controlado para la ejecución de procesos, gracias a los *namespaces* y control *groups* de Linux. Son solo necesarios un pequeño conjunto de ficheros para poder generar el contenedor sobre el sistema operativo. Ejecuta aplicaciones y sus servicios relacionados de una forma aislada e independiente, buscar que se comportase de igual forma en todos los entornos. (Cuervo, 2019)

Los contenedores son escalables, portables y no utiliza una maquina física por cada contenedor, esto ayuda a tener varios contenedores en una sola maquina física para entender mejor se podría interpretar los siguientes conceptos:

Tabla 2. Definición de contenedores

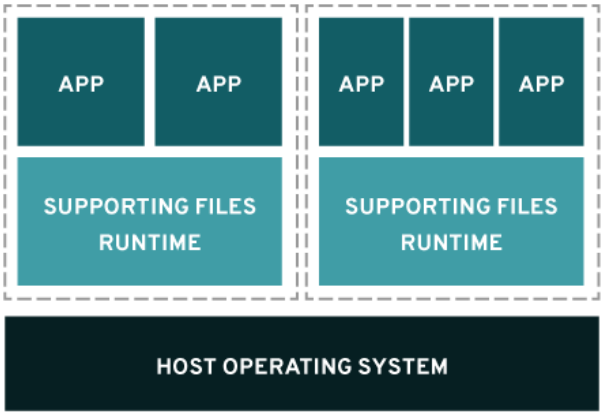
Referencias	Conceptos
(AWS, 2020)	Docker es una plataforma de software que le permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, se podrá implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.
(Rouse, 2019)	La virtualización basada en contenedores aísla las aplicaciones entre sí en un sistema operativo (OS) compartido. Este enfoque estandariza la entrega del programa de la aplicación, permite que las aplicaciones se ejecuten en cualquier entorno Linux, ya sea físico o virtual. Dado que comparten el mismo sistema operativo, los contenedores son portátiles entre diferentes distribuciones de Linux, y son significativamente más pequeños que las imágenes de máquinas virtuales (VM).
(López, 2018, s.p.)	"Docker encapsula todas las dependencias de una aplicación en un contenedor, para conseguir aislar la aplicación y sus dependencias y por tanto obtener una aplicación más fácil de mover entre servidores, sistemas operativos, localizaciones".

Fuente: elaboración propia

Se podría decir que los contenedores son portables y consumen menor recurso que una máquina virtual generara rapidez en la migración de aplicaciones entre servidores o a su vez poner en producción una aplicación que haya pasado por las fases de pruebas. Los contenedores actualmente se utilizan para mejorar la calidad y seguridad de forma escalable, según sea necesario en el ciclo de vida de las aplicaciones.

Un esquema simplificado de su arquitectura se muestra en el Gráfico 4.

Gráfico 4. Esquema de un contenedor



Fuente: (Cuervo, 2019)

CAPÍTULO II. DISEÑO METODOLÓGICO

2.1 Caracterización de la empresa

La seguridad en el ciclo de vida de las aplicaciones cumple un papel importante en el desarrollo integral de aplicaciones por, lo cual, el modelo de estándares para la seguridad en el SDLC ayuda tener aplicaciones funcionales y seguras.

En la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., actualmente no se tiene estándares de seguridad para el monitoreo de las aplicaciones, por lo que se realiza manualmente, esto ocasiona demora y consumo de tiempo en el personal de sistemas de la empresa.

Historia

Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., es una empresa Ambateña-ecuatoriana fundada por: Leonardo Fabio Palate Guaman y Omayra Leticia Quinga Escobar, dedicada a importar y comercializar repuestos para vehículos en distintas líneas tales como: Accesorios, Carrocería, Dirección, Eléctrico, Mantenimiento, Mesa de suspensión, Motor, Refrigeración, Suspensión y transmisión.

La empresa nace en el año 2016 bajo el logo de una imagen del planeta tierra. El logo surge porque la empresa se dedica a importar repuestos para vehículos desde varios países como: China, Japón, México, Brasil entre otros, para distribuir en Ecuador a nivel nacional.

En el año 2016 la compra de los repuestos se lo realizaba nacionalmente y a partir del año 2017 se empezó a importar desde China, Japón y Brasil. Como resultado la expansión de ventas a nivel nacional.

Con el pasar de los años Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., ha tenido mejoras e innovaciones en sus procesos de importación con productos originales y alternos con el fin de ofertar en el mercado nacional repuestos que cumplan con altos estándares de calidad.

El compromiso es comercializar repuestos de excelente calidad para una mejor satisfacción de los clientes.

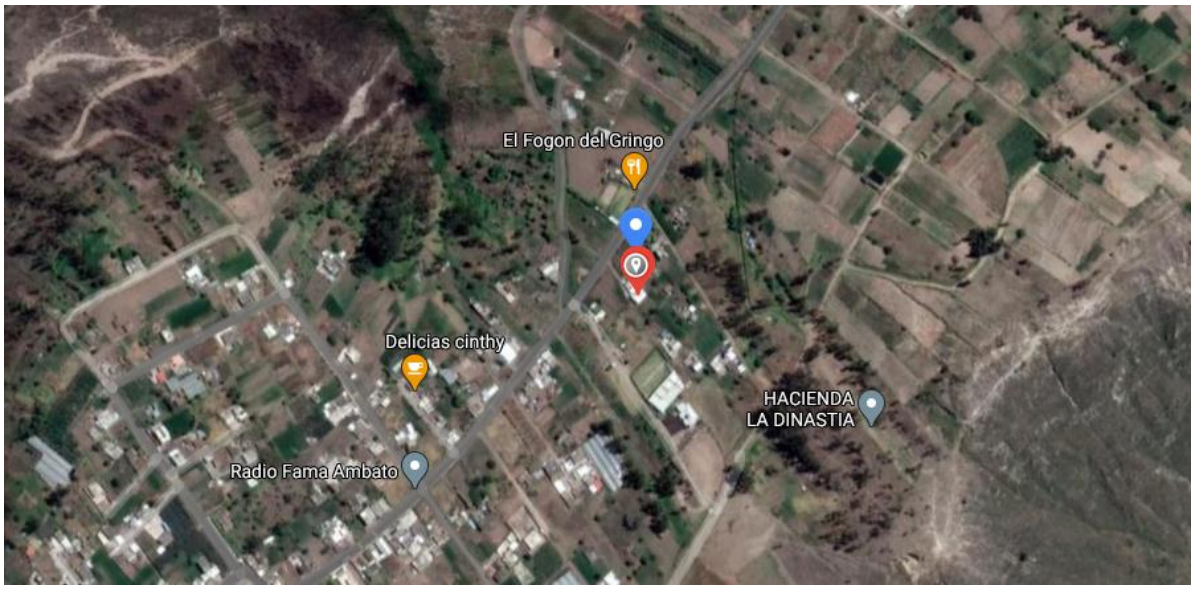
Misión

Somos una empresa dedicada a atender las necesidades de repuestos y servicios en el mercado automotor a través del mejor equipo humano, brinda un excelente servicio de venta, postventa y mantenimiento a nuestros clientes. Somos distribuidores de repuestos originales y alternos de calidad a precios competitivos.

Visión

Ser la mejor empresa importadora y comercializadora de repuestos automotrices en el país, brinda un excelente servicio en ventas y postventa, genera valor agregado para nuestros clientes y la sociedad.

Figura 1. Ubicación de la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda.



Fuente: (Google Maps, 2019)

2.2 Metodología de investigación científica

La metodología de la investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno o problema. Orienta la manera en que va a enfocar una investigación y la forma en que va a recolectar, analizar y clasificar los datos, con el objetivo de que los resultados tengan validez y pertinencia, y cumplan con los estándares de exigencia científica.

“La metodología de la investigación, en este sentido, también, la parte de un proyecto de investigación donde se exponen y describen razonablemente los criterios adoptados en la elección de la metodología” (Coelho, 2019).

Para el presente proyecto se aplicará el método de investigación analítico-sintético que ayuda con la descomposición del objeto de estudio en cada una de sus partes para estudiarlas en forma individual y luego de forma integral, también, es un conjunto de pasos fijados por una disciplina con el fin de alcanzar conocimientos válidos mediante instrumentos confiables, secuencia estándar para formular y responder a una pregunta. El primer paso es examinar en profundidad el objeto de estudio que se tiene entre manos. Para ello, se podría recurrir tanto a la observación directa o a otras técnicas indirectas.

Observación directa

Consiste en saber seleccionar aquello que queremos analizar dentro de la investigación a desarrollar, existe una frase que suele decir que "Saber observar es saber seleccionar" (Palma et al., 2019, p.4). Para la observación lo primero es plantear previamente qué es lo que interesa observar.

> El método de observación directa es un método de recolección de datos que consiste básicamente en observar el objeto de estudio dentro de una situación particular. Todo esto se hace sin necesidad de intervenir o alterar el ambiente en el que se desenvuelve el objeto. De lo contrario, los datos que se obtengan no van a ser válidos. (OKDIARIO, 2019)

En sentido general, la observación se podría considerar como un objeto observado donde se desenvuelve sin que moleste al observador. Por todo ello, los datos obtenidos a través de este método son reconocidos y tienen renombre en el área de la investigación, en el (Anexo 1) permite determinar el registro de datos observados para posterior análisis e interpretación.

Resultado de la ficha de observación

Tabla 3. Ficha de observación del monitoreo de aplicaciones

FICHA DE OBSERVACIÓN DIRECTA	
Ficha N.:	01
Investigador:	Fredy Leonardo Arroba Flores
Motivo para observar:	Obtener información sobre el monitoreo de las aplicaciones.
Observaciones: La presente observación es para constatar el monitoreo de las aplicaciones en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., proceso actual para el monitoreo es de forma manual esto dificulta la toma de decisiones de forma rápida y oportuna para resolver vulnerabilidades que se encuentran dentro de ciclo de vida las aplicaciones. Los pasos para el monitoreo de las aplicaciones actualmente son las siguientes: 1. En el servidor se busca el log por fecha. 2. Abrir el log con un editor de texto. 3. Buscar errores o alertar que se hayan ejecutado. 4. Comparar los errores o alertas con otras similares para hacer un análisis concurrente de estos procesos que se ejecutan. 5. Tomar decisiones y corregir en el código fuente o en la base de datos según sea el caso de análisis. Todo este proceso es por fecha, para un análisis entre varias fechas se vuelve casi imposible ejecutar al 100% todos los logs que generan en las aplicaciones.	

Fuente: elaboración propia

Tabla 4. Ficha de observación de las seguridades en el ciclo de vida de las aplicaciones

FICHA DE OBSERVACIÓN DIRECTA	
Ficha N.:	02
Investigador:	Fredy Leonardo Arroba Flores
Motivo para observar:	Obtener información sobre la seguridad en ciclo de vida de las aplicaciones
Observaciones: Constatar la seguridad en el ciclo de vida de las aplicaciones en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., actualmente no existe estándares de seguridad en las aplicaciones que se encuentran implementadas, por lo que el monitoreo es manual y consumen tiempo operativo en el área de sistemas. Los desarrollos de nuevas aplicaciones o mejoras en las existentes se lo realizan bajo demanda, según las necesidades de los usuarios finales, por lo que no se tiene estándares en el ciclo de vida de las aplicaciones lo que ocasiona en los desarrollos tarden más de lo previsto. La seguridad en las aplicaciones actualmente no se aplica al 100%, porque se necesita de un monitoreo de todas las aplicaciones para corregir las vulnerabilidades y tener segura la información.	

Fuente: elaboración propia

De acuerdo con la observación realizada en el área de sistemas, se podría manifestar que las aplicaciones no tienen SDLC, por lo cual, la información alojada en el servidor On-premise (servidor local) no se encuentra segura. El monitoreo de las aplicaciones que se encuentran en

producción y en desarrollo se lo realizan manualmente, lo cual, consume tiempo en el departamento de sistemas.

Se podrá evidenciar que las aplicaciones son lentas en el procesamiento de la información, no existe un control adecuado en el ciclo de vida de las aplicaciones.

Entrevistas

Las entrevistas es una forma específica de interacción social que tiene como meta obtener datos para la investigación. Resultan de utilidad si el tema de la investigación se relaciona con asuntos que requieren sondeo considerable y cuestionamiento complejo. Las entrevistas según Rodriguez (2020) son conversaciones que se llevan a cabo para recoger información, esta técnica en investigación involucra a un entrevistador, quien es el que dirige el proceso y hace las preguntas, y un entrevistado que responde. Las entrevistas podrán realizarse cara a cara, telefónicamente o por medios virtuales (correo electrónico, formulario en línea, por vídeo conferencia).

A fin de sustentar la investigación se utiliza la entrevista (Anexo 2) como respectivo instrumento; para la entrevista se desarrolla dentro de la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., con la finalidad de obtener información acerca del monitoreo en el ciclo de vida de las aplicaciones, la cual, se aplicó la entrevista semiestructurada para la obtención de información que aporta en el desarrollo del proyecto, en cuanto a la entrevista está orientada al departamento de sistemas conformada por 2 personas, quienes son los responsables directos de monitoreo y seguridades en el ciclo de vida de las aplicaciones .

Resultados de la Entrevista

Tabla 5. Resumen de la entrevista para conocer importancia del monitoreo de aplicaciones

1. ¿Cuán importante es realizar el monitoreo de las aplicaciones?
• **R1:** El monitoreo de las aplicaciones es importante, se podrá identificar vulnerabilidades en el código mediante el monitoreo de los logs. • **R2**: Actualmente el monitoreo de los sistemas no está enfocada a la seguridad dentro del ciclo de vida de las aplicaciones, lo cual, afecta directamente a la seguridad de la información.
Análisis: En la actualidad, el monitoreo de los sistemas está enfocada al funcionamiento correcto de las aplicaciones, por lo que provocan vulnerabilidades para posibles ataques.

Fuente: elaboración propia

Tabla 6. Resumen de la entrevista para conocer tiempo dedicado en el monitoreo de SDLC

2. ¿Cada que tiempo se dedica a realizar el monitoreo de las aplicaciones?
• **R1:** El monitoreo de las aplicaciones se lo realiza en la última semana de cada mes. • **R2**: El monitoreo se lo realiza cada mes, pero si el sistema esta lento o en el firewall detecta ataques, se monitorea en ese momento.
Análisis: El monitoreo se lo realiza mensualmente, el tiempo asignado para el monitoreo es amplio, en el transcurso de este intervalo podría existir ataques a las aplicaciones y la información pasa a un plano de vulnerabilidad.

Fuente: elaboración propia

Tabla 7. Resumen de la entrevista para conocer estándares de seguridad en las aplicaciones

3. ¿Tiene estándares de seguridad para el monitoreo de las aplicaciones?
• **R1:** Actualmente no se aplica estándares de seguridad en el monitoreo de las aplicaciones. • **R2**: No existe estándares de seguridad para el monitoreo de las aplicaciones, solo se monitorea la funcionalidad de los sistemas.
Análisis: El monitoreo de las aplicaciones está enfocado a la funcionalidad mientras que la seguridad no es tomada en cuenta, las aplicaciones quedan vulnerables a posibles ataques.

Fuente: elaboración propia

Tabla 8. Resumen de la entrevista para conocer el monitoreo de los logs en las aplicaciones

4. ¿Cómo realiza el monitoreo de los logs en las aplicaciones?
• **R1:** El monitoreo lo realiza manualmente, se ingresa al directorio donde están los logs de las aplicaciones, para luego abrir el log actual y proceder a revisar. • **R2**: El monitoreo de logs es un proceso que se lo realiza manualmente, es abrir el log actual con un editor de texto, para luego proceder analizar los diferentes eventos que contiene el log.
Análisis: En la actualidad el monitoreo de los logs en las aplicaciones se lo realiza de forma manual, lo cual, consume tiempo en el analizar de los diferentes eventos generados en los logs.

Fuente: elaboración propia

Tabla 9. Resumen de la entrevista para conocer herramientas de monitoreo en las aplicaciones

5. ¿Aplica alguna herramienta para monitoreo de las aplicaciones?
• **R1:** Actualmente no se utiliza ninguna herramienta que ayude al monitoreo de las aplicaciones. • **R2**: En el monitoreo de las aplicaciones no se aplica ninguna herramienta que ayude con la automatización al momento de revisar las aplicaciones.
Análisis: Actualmente no existe ninguna herramienta que ayude con la automatización al momento del monitoreo de las aplicaciones, lo cual, consume tiempo en el análisis de las aplicaciones.

Fuente: elaboración propia

Tabla 10. Resumen de la entrevista para conocer el monitoreo en el SDLC

6. ¿Realiza monitoreo en ciclo de vida de las aplicaciones?
• **R1:** Si existe el monitoreo en el ciclo de vida de las aplicaciones desde el diseño de la aplicación hasta puesta en producción, pero solo está enfocado a la funcionalidad. • **R2**: Actualmente se realiza el monitoreo en el ciclo de vida de las aplicaciones tanto los sistemas que están en producción como nuevos desarrollos.
Análisis: El monitoreo en el ciclo de vida de las aplicaciones si se lo realiza actualmente, pero solo enfocado a la funcionalidad, mientras que la seguridad de la información no se le lo toma en cuenta.

Fuente: elaboración propia

En la actualidad el monitoreo en el ciclo de vida de las aplicaciones no se aplican seguridades por lo que solo están enfocados a la funcionalidad y la seguridad queda en segundo plano, esto provoca que las aplicaciones tengan vulnerabilidades y la información no está segura. El monitoreo tiene que ser fácil de realizar y detectar ataques de forma que se podrá tomar decisiones de forma oportuna.

Población y Muestra

La población lo conforman: el proveedor del sistema que es externo y el asistente de sistemas que trabaja en el departamento de sistemas dentro de la empresa. Por lo siguiente no se requiere la realización del cálculo de muestreo porque la población es pequeña.

2.3 Metodología de desarrollo

La metodología para utilizar en el proyecto es Kanban, sirve para visualizar el trabajo, evitar la acumulación de trabajo pendiente y maximizar la eficiencia. Es un proceso que permite mejorar constantemente el flujo y la calidad del trabajo. Según Atlassian (2019) consiste en visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia (o el flujo). Los equipos de Kanban se centran en reducir el tiempo que se tarda en llevar un proyecto (o historia de usuario) del principio al fin del proceso. Para ello, utilizan un Kanban y mejoran continuamente su flujo de trabajo.

2.3.1 Características de la metodología Kanban

Según Kanbanize (2018) las características la metodología Kanban son las siguientes:

- **Visualizar el flujo de trabajo**

  Lo primero y lo más importante es entender qué se necesita para el transcurso de un producto desde su pedido hasta su entrega.

- **Eliminar las interrupciones**

  El cambio de enfoque podría dañar seriamente su proceso y la multitarea (o multitasking) podría provocar generación de desperdicios.

- **Gestionar el flujo**

  La idea de implementar un sistema Kanban es crear un flujo continuo e ininterrumpido. Por flujo nos referimos al movimiento de elementos de trabajo a través del proceso de producción. Lo que interesa es la velocidad y la continuidad del movimiento.

- **Hacer las políticas explícitas (Fomentar la visibilidad)**

  No se podría mejorar algo que no se entiende. Esta es la razón, la cual, el proceso estará bien definido, publicado y promovido. Las personas no se asociarían ni participarían en algo que no creen que sea útil.

- **Circuitos de retroalimentación**

Para que el cambio positivo ocurra, tenga éxito y sea duradero, se necesita hacer una cosa más. La filosofía Lean admite que las reuniones regulares son necesarias para la transferencia de conocimiento (circuitos de retroalimentación).

- **Mejorar colaboración (usa modelos y el método científico)**

La forma de lograr la mejora continua y el cambio sostenible dentro de una organización se consigue a través de la visión compartida para un futuro mejor y la comprensión colectiva de los problemas que superen.

Para aplicar los estándares de seguridad en las aplicaciones se implementa la metodología de desarrollo, seguir paso a paso las tareas establecidas en el proyecto.

La metodología de desarrollo ayuda a organizar las tareas a seguir, con el objetivo de evidenciar la seguridad en los procesos del ciclo de vida de las aplicaciones de la empresa.

2.3.2 Fases de la metodología Kanban

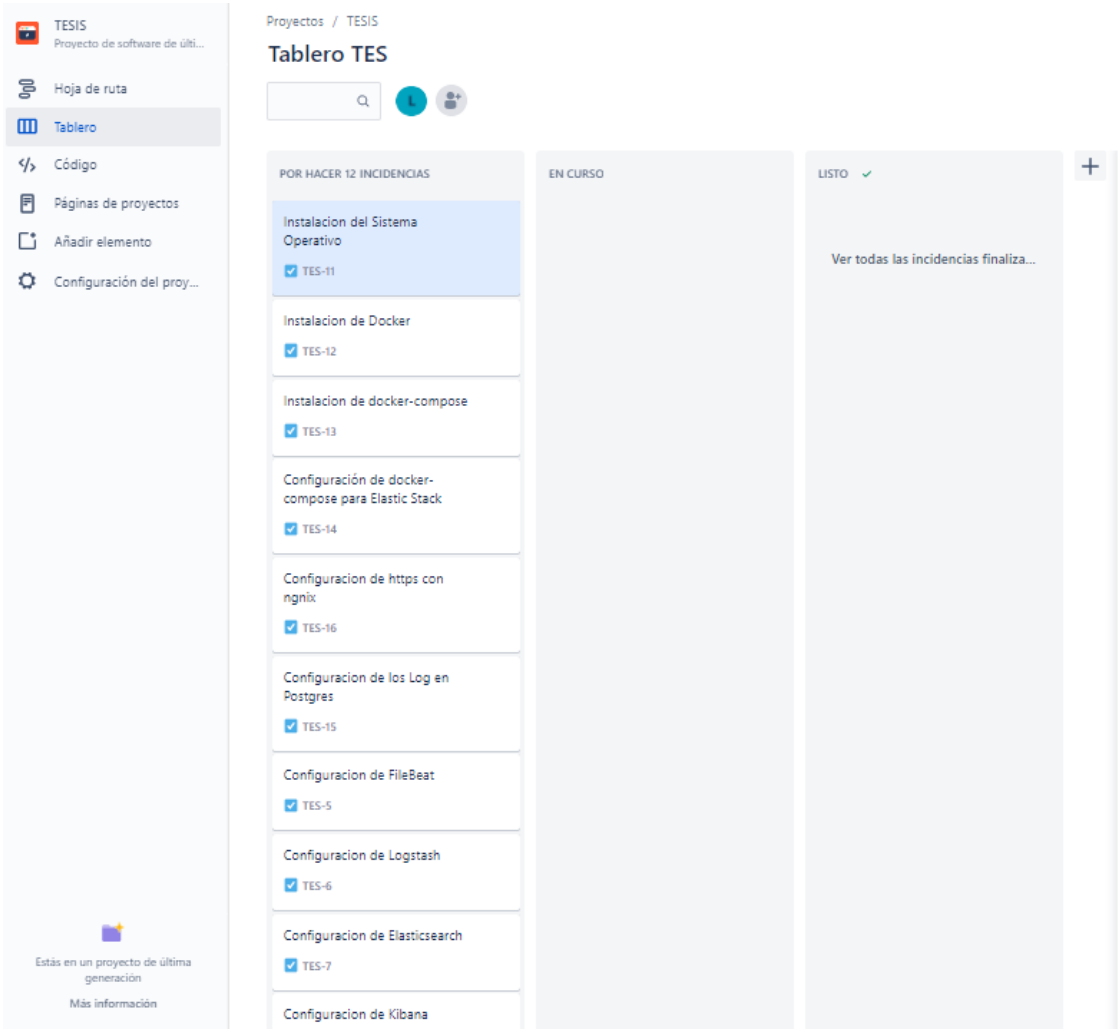
La metodología Kanban se basa en un tablero cuyo objetivo es gestionar de manera general cómo se completan las tareas. Las tareas pasan por cada flujo de trabajo con se muestra en la Figura 2: petición de tareas, selección de tareas, desarrollo, pruebas y terminado con el objetivo de cumplir cada tarea asignada, a continuación, se listan las tareas que se utilizaron para el monitoreo en el ciclo de vida de las aplicaciones.

Figura 2. Tablero Kanban

Fuente: (Toledo Garcia, 2017)

Para trabajar con la metodología Kanban se utilizará la herramienta Jira. Regina (2018) menciona que es una herramienta que ayuda al equipo a gestionar sus tareas y actividades. Reside en seguir los principios de Agile tanto dentro de la empresa como hacia afuera con la comunidad que la rodea. En la Figura 3, se muestran todas las tareas que se van a realizar en el presente proyecto.

Figura 3. Tareas en Jira



Fuente: elaboración propia

Para una mejor explicación se detalla los objetivos por cada tarea a desarrollar, como se muestra en la Tabla 11.

Tabla 11. Objetivos de tareas a desarrollar

Tareas	Objetivos
Instalación del Sistema Operativo	Tener un sistema operativo para instalar y configurar Elastic Stack.
Instalación de *Docker*	Permite la creación de paquetes que incluye todo lo necesario para que Elastic Stack funcione correctamente.
Instalación de *docker-compose*	Ejecutar varios contenedores de Docker, se utiliza un archivo en el formato yml, en el que se define las configuraciones.
Configuración de *docker-compose* para Elastic Stack	Parametrizar el archivo de configuración para que el despliegue de Elastic Stack.
Configuración de https con *ngnix*	Tener certificado de seguridad para la transferencia de datos cifrados en un navegador y un servidor web.
Configuración de los *Log* en Postgres	Configurar para que se genere los logs con errores y advertencia en un solo archivo.
Configuración de *FileBeat*	Parametrizar la ubicación donde se generan los logs y enviar a *logstash.*
Configuración de *Logstash*	Depurar la información de los logs y enviar a *Elasticsearch* datos limpios.
Configuración de *Elasticsearch*	Almacenar los logs mediante índices.
Configuración de *Kibana*	Construir tableros con informes dinámicos para tomar decisiones ante la detección de vulnerabilidades en el sistema.

Fuente: elaboración propia

Tarea 1. Instalación del sistema Operativo

Antes de iniciar con la instalación, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 4.

Figura 4. Tarea, instalación del sistema operativo en curso

Fuente: elaboración propia

Para este proyecto se instala el sistema operativo Centos versión 7 en un servidor *on-premise* (infraestructura física o local), es un sistema operativo escalable, estable y uno de las más seguros en ambientes Linux y soporta a la perfección Docker donde se va a implementar *Elastic Stack.*

> Centos, viene del inglés, *Community Enterprise Operating System*, hace una traducción libre en español algo así como Sistema Operativo para la Comunidad Empresarial. CentoS es una distribución particular (también, conocida como distro) del sistema operativo Linux. Conocida por su estabilidad, consistencia, administración fácil de usar y replicación directa, esta versión del sistema operativo de código abierto se creó como una escisión de Red Hat Enterprise Linux (RHEL). (Centos, 2018)

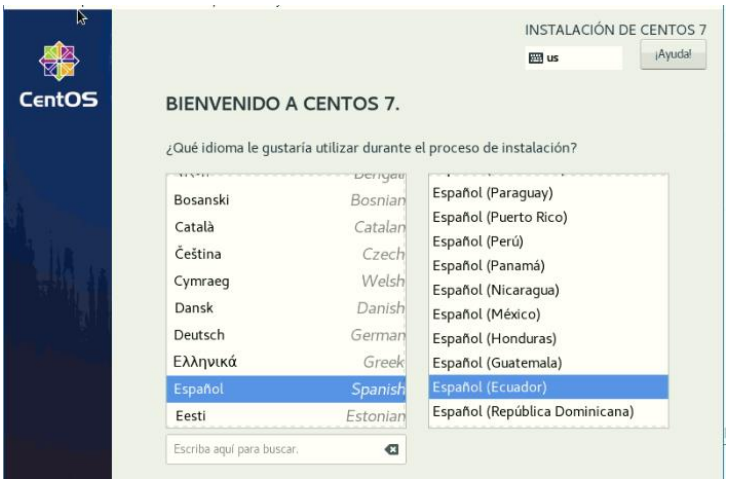
Para comenzar con la instalación del sistema operativo, el primer paso es descargar, para ello se ingresó a la siguiente página: http://mirror.cedia.org.ec/centos/7/isos/x86_64/ para descarga la imagen del sistema operativo.

Los requisitos indispensables para la instalación del sistema operativo son los siguientes:

- RAM de 8GB.
- 100 GB espacio en disco.
- Conectividad a internet.

Una vez descargado el instalador y cumplir con las características y requisitos, se procede a la instalación como se muestra en la en la Figura 5.

Figura 5. Seleccionar Idioma Centos 7



Fuente: elaboración propia

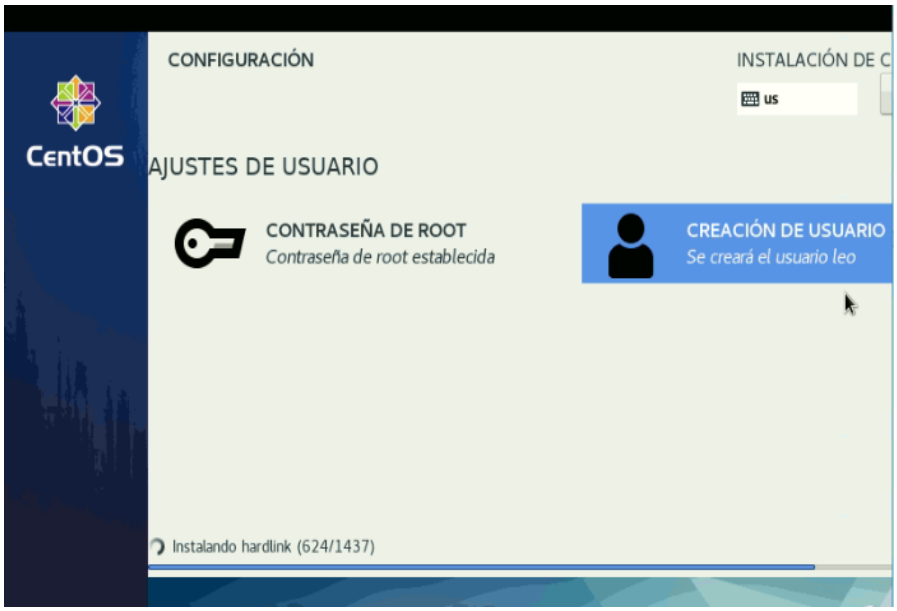
Se procede a la configuración de las siguientes opciones como se muestra en la Figura 6.

Figura 6. Configuración Centos

Fuente: elaboración propia

Se habilitará el botón **empezar la instalación** si todo está configurado correctamente, dar *click* en el botón; a continuación, se desplegará una ventana para crear un usuario y establecer la contraseña para el usuario *root* (super usuario) como se muestra en la en la Figura 7.

Figura 7. Creación de usuario y contraseña para usuario *root*



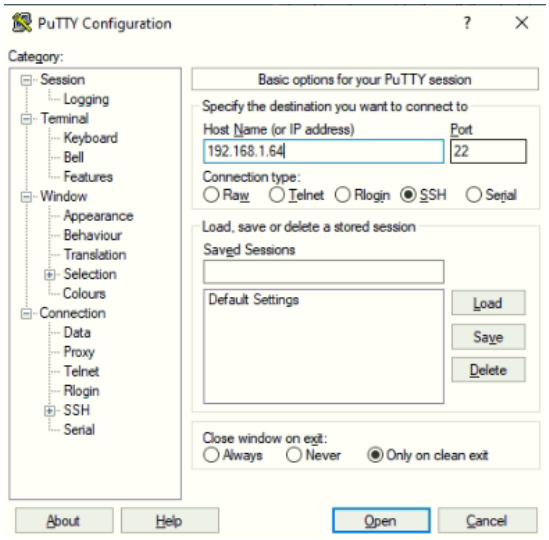
Fuente: elaboración propia

Una vez completada la instalación se podrá acceder mediante SSH (*Secure Shell* o intérprete de órdenes seguro), que sirve para acceder a máquinas remotas a través de una red.

SSH: Permite manejar por completo un servidor mediante un intérprete de comandos. Hostinger (2017) menciona que es un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada.

Para usar el protocolo SSH se utilizará el cliente *Putty*, la cual, permite conectar al servidor con el protocolo ssh como se muestra en la Figura 8.

Figura 8. Conectar al servidor con la herramienta Putty



Fuente: elaboración propia

Para finalizar la tarea de la instalación del sistema operativo, en el tablero de Kanban se mueve la tarea al estado **listo** como se muestra en la Figura 9.

Figura 9. Tarea para instalación del sistema operativo estado listo

Fuente: elaboración propia

Tarea 2. Instalación de Docker

Antes de iniciar con la instalación, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 10.

Figura 10. Tarea para instalación de Docker, estado en curso

Fuente: elaboración propia

Para la instalación de *docker* se despliega paquetes adiciones, para que la ejecución sea satisfactoria. La instalación estará con el usuario *root* y ejecutar la siguiente línea de comando como se muestra en la Figura 11:

Figura 11. Instalación de paquetes para Docker

Fuente: elaboración propia

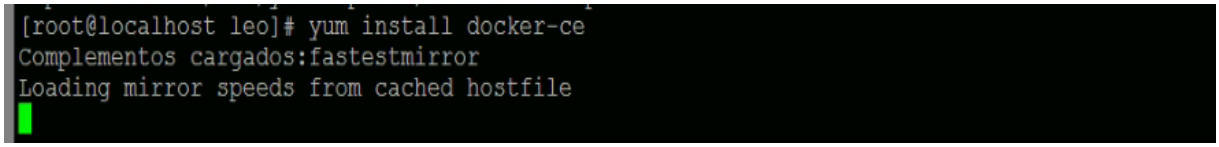
Para descargar el instalador de Docker la forma más fácil y segura de completar el proceso es a través de los repositorios oficiales de Docker. Para hacer esto, es necesario ejecutar la siguiente línea de comando como se muestra en la Figura 12:

Figura 12. Descarga de Docker

Fuente: elaboración propia

Después de la descarga, se procede a instalar Docker, ejecutar el siguiente comando como se muestra en la Figura 13:

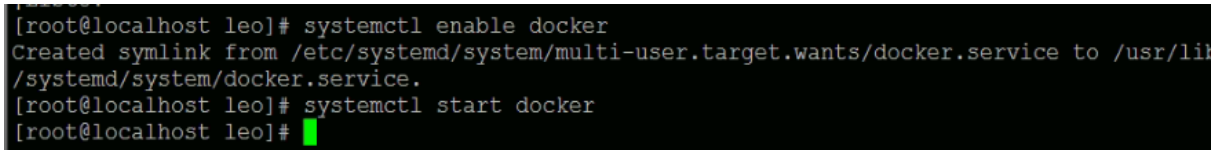
Figura 13. Instalación de Docker



Fuente: elaboración propia

Terminada la instalación de Docker en Centos 7, el paso siguiente es habilitar e iniciar el servicio de Docker ejecutar el siguiente comando como se muestra en la Figura 14:

Figura 14. Iniciar el servicio de Docker



Fuente: elaboración propia

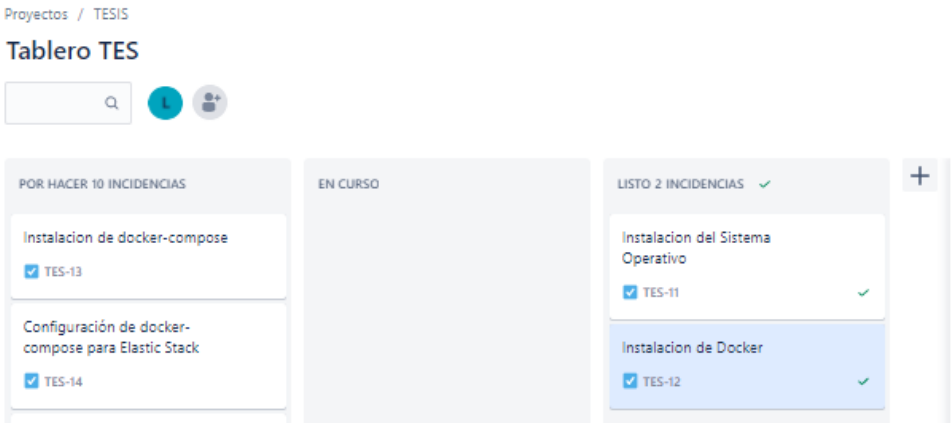
Finalmente, verificar el estado del servicio para chequear que todo haya salido bien ejecutar el siguiente comando como se muestra en la Figura 15:

Figura 15. Verificar el estado del servicio de Docker

Fuente: elaboración propia

Para finalizar la tarea de la instalación de *docker*, en el tablero de Kanban se mueve la tarea al estado **en listo** como se muestra en la Figura 16.

Figura 16. Tarea para instalación de Docker, estado listo

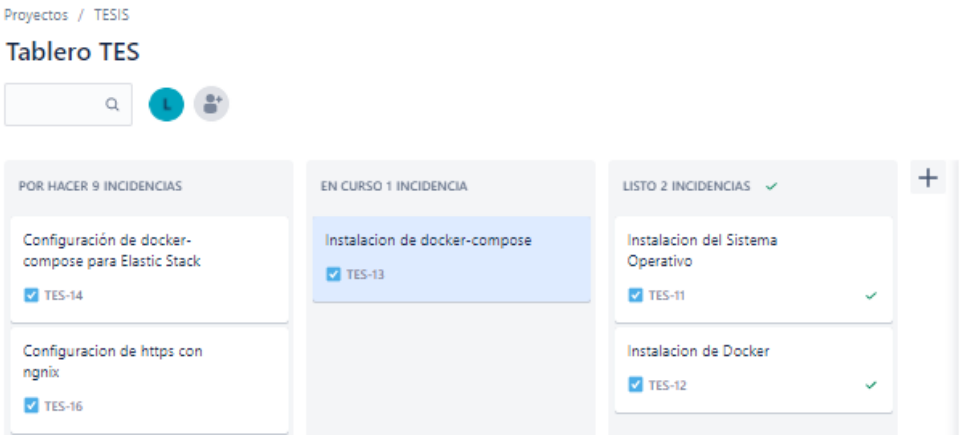


Fuente: elaboración propia

Tarea 3. Instalación de Docker-Compose

Antes de iniciar con la instalación, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 17.

Figura 17. Tarea para instalación de *docker-compose*, estado en curso



Fuente: elaboración propia

Compose es una herramienta para definir y ejecutar aplicaciones *Docker* de varios contenedores. Con *Compose*, se utiliza un archivo YAML (no es un lenguaje de marcado) para configurar los servicios de su aplicación. Luego, con un solo comando, se crea e inicia todos los servicios desde su configuración. *Compose* funciona en todos los entornos: producción, puesta en escena, desarrollo, pruebas y flujos de trabajo de CI (Integración Continua). (Docker-Compose, 2021).

Para instalar Docker Compose en CentOS 7, necesitamos ejecutar el siguiente comando como se muestra en la Figura 18.

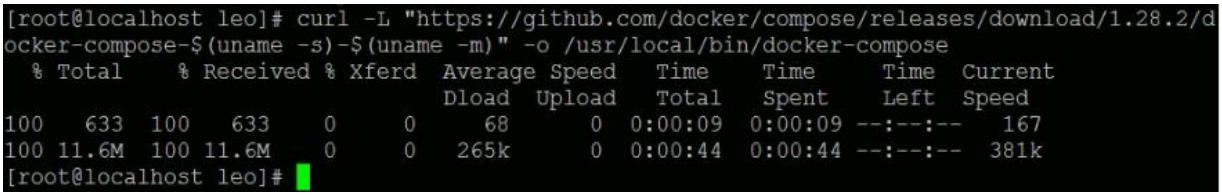
Figura 18. Instalación de *docker-compose*

```
[root@localhost leo]# curl -L "https://github.com/docker/compose/releases/download/1.28.2/d
ocker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633  100   633    0     0     68      0  0:00:09  0:00:09 --:--:--   167
100 11.6M  100 11.6M    0     0   265k      0  0:00:44  0:00:44 --:--:--  381k
[root@localhost leo]#
```

Fuente: elaboración propia

Una vez instalado se procede asignar permisos de ejecución al *docker-compose* binario necesitamos ejecutar el siguiente comando como se muestra en la Figura 19:

Figura 19. Asignar permisos de ejecución a *docker-compose*



Fuente: elaboración propia

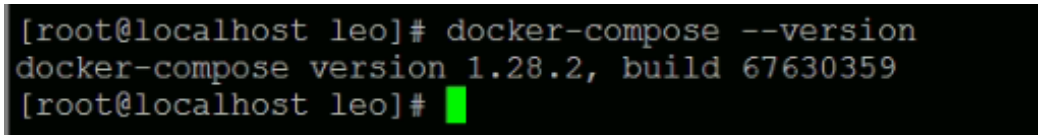
Para garantizar que no haya problemas al usar la herramienta en el terminal, se realizara un enlace simbólico al sistema se ejecutara el siguiente comando como se muestra en la Figura 20:

Figura 20. Enlace simbólico a *docker-compose*

Fuente: elaboración propia

Finalmente, verificar la versión instalada, se ejecutará el siguiente comando como se muestra en la Figura 21:

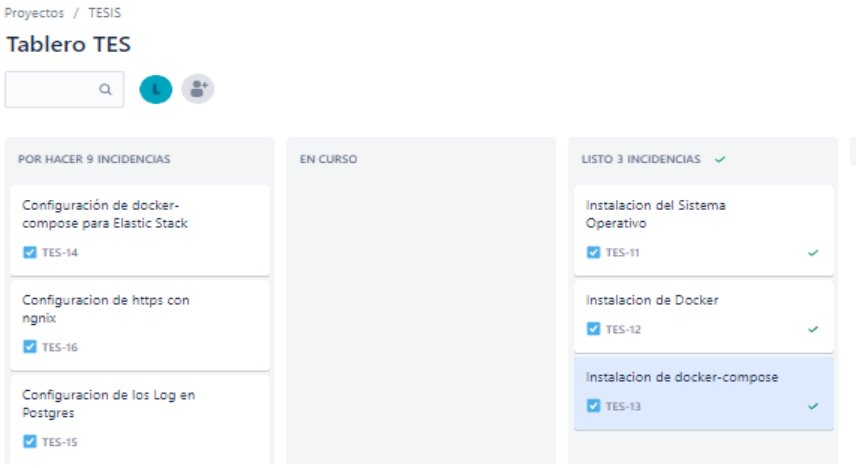
Figura 21. Verificar *docker-compose*



Fuente: elaboración propia

Para finalizar la tarea de la instalación de *Docker-compose*, en el tablero de Kanban se mueve la tarea al estado **en listo** como se muestra en la Figura 22.

Figura 22. Tarea para instalación de *docker-compose*, estado listo

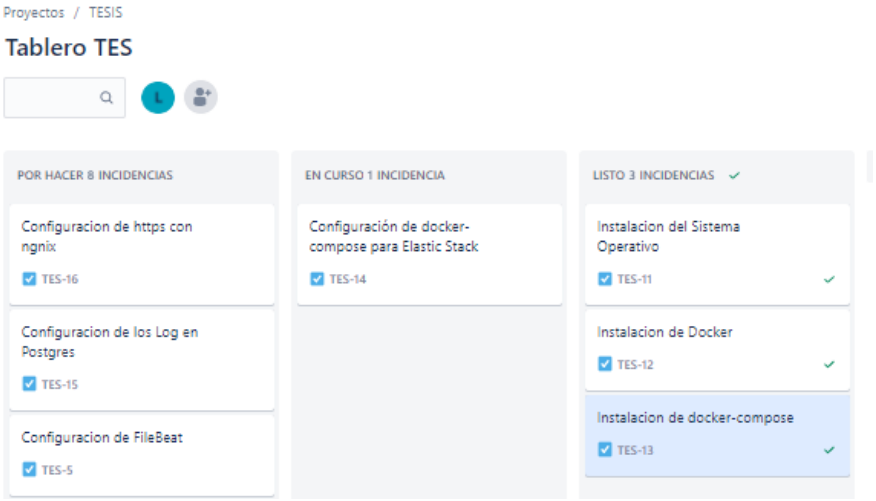


Fuente: elaboración propia

Tarea 4. Configuración de docker-compose para Elastic Stack

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 23.

Figura 23. Tarea para configuración de *docker-compose*, estado en curso



Fuente: elaboración propia

Elastic Stack o ELK Stack es un conjunto de herramientas *open source* desarrolladas por Elastics que permite recoger datos de cualquier tipo de fuente y en cualquier formato para realizar búsquedas, análisis y visualización de los datos en tiempo real. Según García (2019), permite crear un sistema de monitorización para la extracción, almacenamiento y explotación de los datos de los procesos o sistemas a monitorizar. Permite la obtención de datos a partir de ficheros de logs mediante *Logstash* y su almacenamiento en el motor de búsquedas y análisis *Elasticsearch*. Además, permite la visualización, monitorización y explotación de datos en tiempo real mediante *Kibana* y la configuración de alertas con *ElastAlert*.

Para la instalación se utilizará Docker-Compose, como primer paso clonar los archivos que están alojados en la nube de forma pública con el objetivo de tener en el servidor para esto se ejecutará el siguiente comando como se muestra en la Figura 24.

Figura 24. Clonación de repositorio *Elastic Stack*

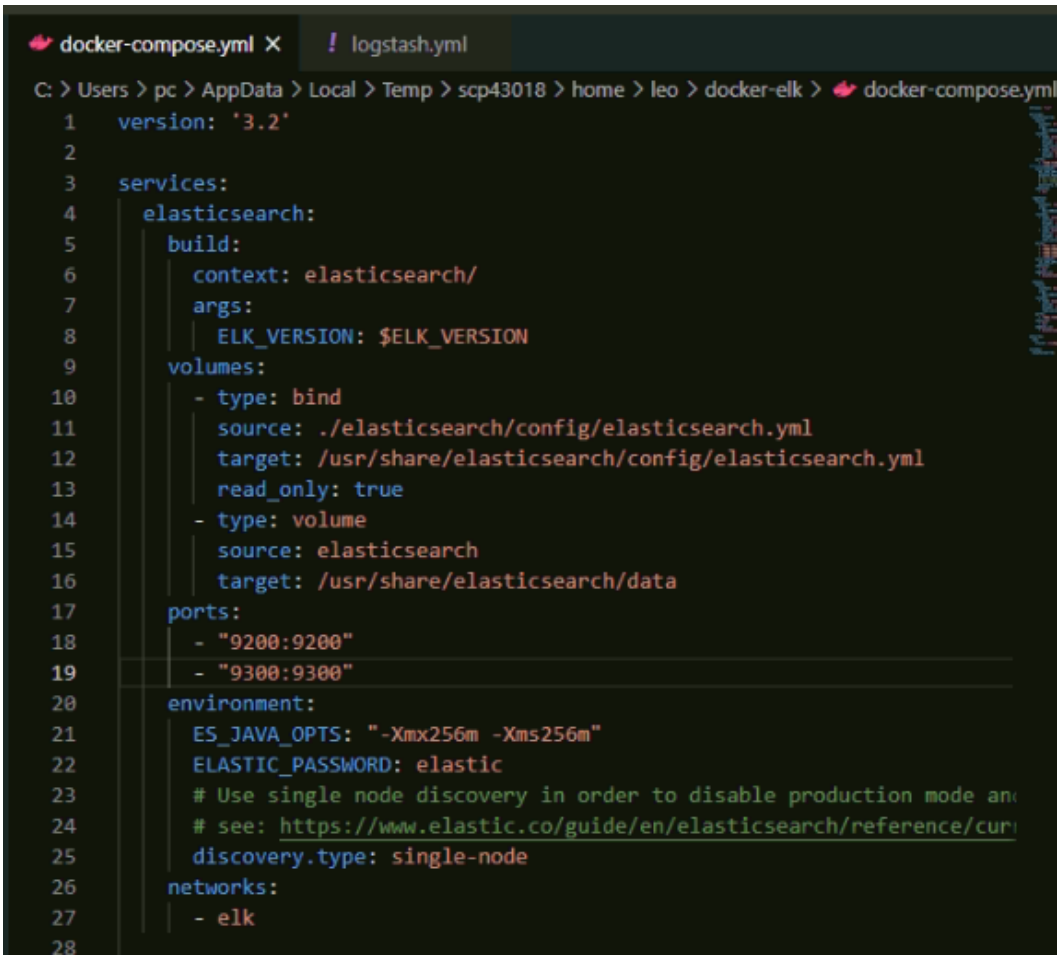
Fuente: elaboración propia

Una vez clonado el repositorio, parametrizar la versión que se va a instalar como se muestra, a continuación: $ELK_VERSION = 7.11.2

Se procederá a la configuración del archivo **docker-compose.yml**, donde están las configuraciones de la instalación con las siguientes herramientas que se utilizaran para el despliegue de *Elastic Stack*, que se detalla a continuación:

ElasticSearch: Es un motor de analítica y análisis distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados. Para el presente proyecto se procederá con la siguiente configuración como se muestra en la Figura 25.

Figura 25. Configuración de la instalación para *ElasticSearch*



Fuente: elaboración propia

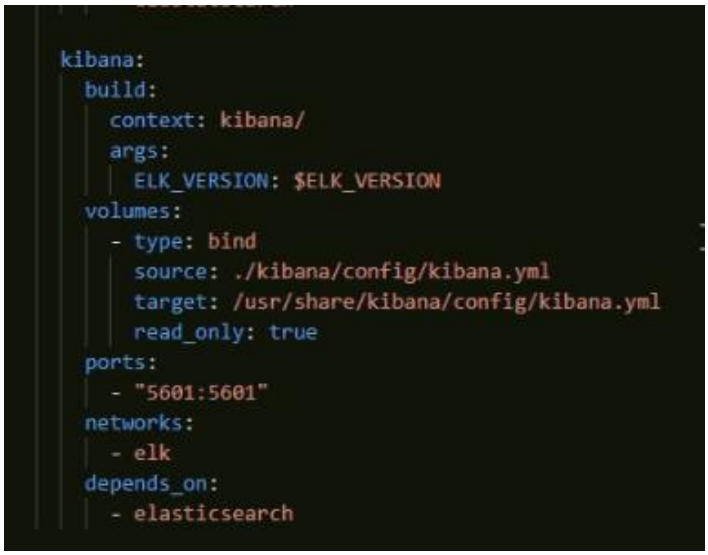
***Logstash*:** Es un procesador de datos de código abierto, las principales funciones son: ingesta, transforma y envía de forma dinámica los datos independientemente de su formato o complejidad. Para el presente proyecto se procederá con la siguiente configuración como se muestra en la Figura 26.

Figura 26. Configuración de la instalación para *Logstash*

Fuente: elaboración propia

***Kibana*:** Es una interfaz web escalable para la representación visual de datos, también, ofrece análisis automáticos en tiempo real, un algoritmo de búsqueda muy flexible y diferentes tipos de vistas (histogramas, gráficos, diagramas circulares, etc.) para los datos. Para el presente proyecto se procederá con la siguiente configuración como se muestra en la Figura 27.

Figura 27. Configuración de la instalación para *Kibana*



Fuente: elaboración propia

Una vez configurado el archivo docker-compose.yml, se procede a desplegar *Elastic Stack* con el siguiente comando como se muestra en la Figura 28.

Figura 28. Despliegue de *Elastic Stack*

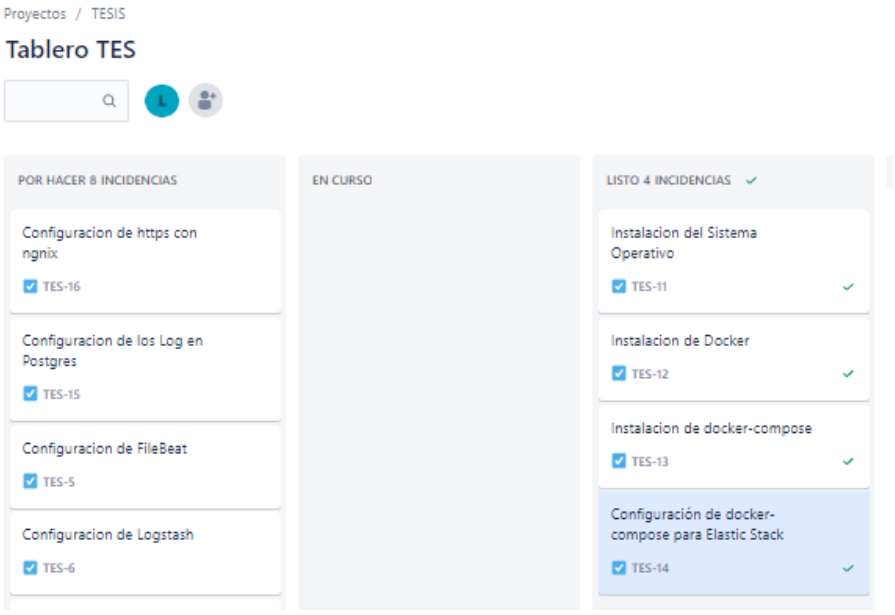
```
[leo@localhost docker-elk]$ sudo docker-compose up -d
[sudo] password for leo:
Building with native build. Learn about native build in Compose here: https://docs.docker.c
om/go/compose-native-build/
Creating network "docker-elk_elk" with driver "bridge"
Creating volume "docker-elk_elasticsearch" with default driver
Building elasticsearch
Sending build context to Docker daemon  3.584kB

Step 1/2 : ARG ELK_VERSION
Step 2/2 : FROM docker.elastic.co/elasticsearch/elasticsearch:${ELK_VERSION}
```

Fuente: elaboración propia

Para finalizar la tarea de la configuración de *docker-compose*, en el tablero de Kanban se mueve la tarea al estado **en listo** como se muestra en la Figura 29.

Figura 29. Tarea para configuración de *docker-compose*, estado listo

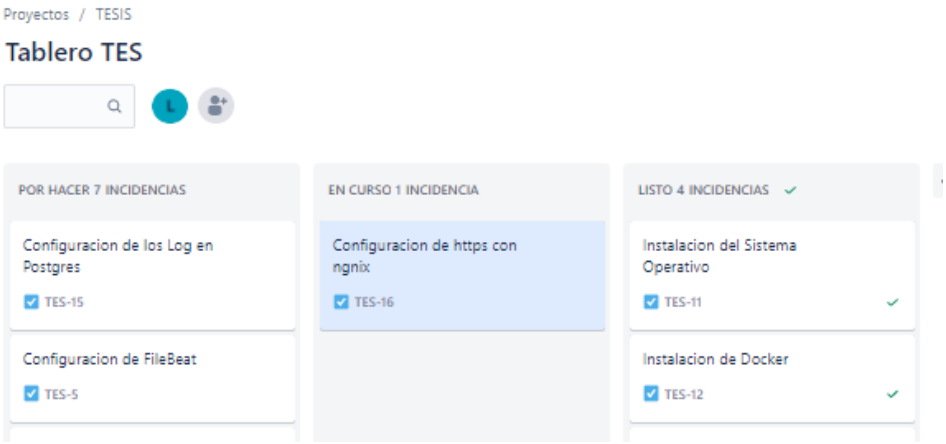


Fuente: elaboración propia

Tarea 5. Configuración de https con ngnix

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 30.

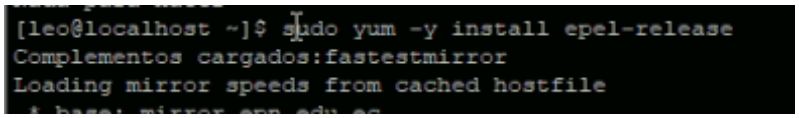
Figura 30. Tarea para la configuración de https con *nginx*, estado en curso



Fuente: elaboración propia

Para instalar *nginx* en Centos 7, se procede a la instalación del repositorio EPEL, como se muestra en la Figura 31.

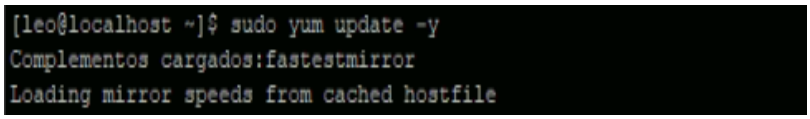
Figura 31. Instalación del repositorio EPEL



Fuente: elaboración propia

Una vez instalado EPEL y configurado en el sistema, actualizar la información de los repositorios y los paquetes instalados, como se muestra en la Figura 32.

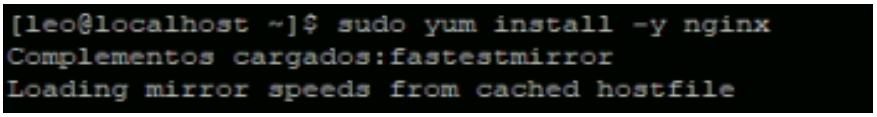
Figura 32. Actualizar el sistema operativo



Fuente: elaboración propia

El siguiente paso es descarga e instalar el paquete *nginx*, como se muestra en la Figura 33.

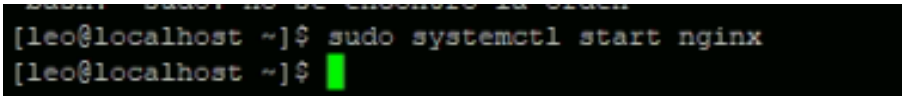
Figura 33. Instalación del paquete *nginx*



Fuente: elaboración propia

Finalizada la instalación de *nginx* y sus dependencias, se habilita el servicio para que arranque automáticamente, como se muestra en la Figura 34.

Figura 34. Habilitar servicio de *nginx*



Fuente: elaboración propia

Para validar que la instalación y configuración del paquete *nginx* este correcto, se valida con el siguiente comando, como se muestra en la Figura 35.

Figura 35. Validar el servicio de *nginx*

Fuente: elaboración propia

Para acceder remotamente a través de la *IP* del servidor, se añade excepciones que permitan conectar desde otras máquinas al servidor *nginx* utilizar *firewall-cmd*, como se muestra en la Figura 36.

Figura 36. Anadir excepciones en el firewall para el servicio de *nginx*

```
[leo@localhost ~]$ sudo firewall-cmd --permanent --add-service={http,https}
success
[leo@localhost ~]$
```

Fuente: elaboración propia

Una vez agregado las excepciones en el *firewall*. Se carga las reglas del *firewall* para que las nuevas excepciones tengan efecto, como se muestra en la Figura 37.

Figura 37. Anadir excepciones en el *firewall* para el servicio de *nginx*

```
[leo@localhost ~]$ sudo firewall-cmd --permanent --add-service={http,https}
success
[leo@localhost ~]$
```

Fuente: elaboración propia

Validar en un navegador que el servidor web de *nginx* esté operativo en producción correctamente, como se muestra en la Figura 38.

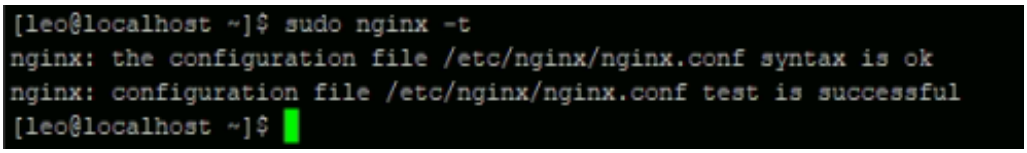
Figura 38. Validar en un navegador el servicio de *nginx*

Fuente: elaboración propia

Para configurar nginx, hay que tener en cuenta que todos los archivos de configuración se encuentran en la siguiente ruta: /etc/nginx/, es el archivo principal de configuración /etc/nginx/nginx.conf.

Cada vez que se realiza cambios en los archivos de configuración de *nginx*, se procede a testear la configuración que este correcta con el siguiente comando, como se muestra en la Figura 39.

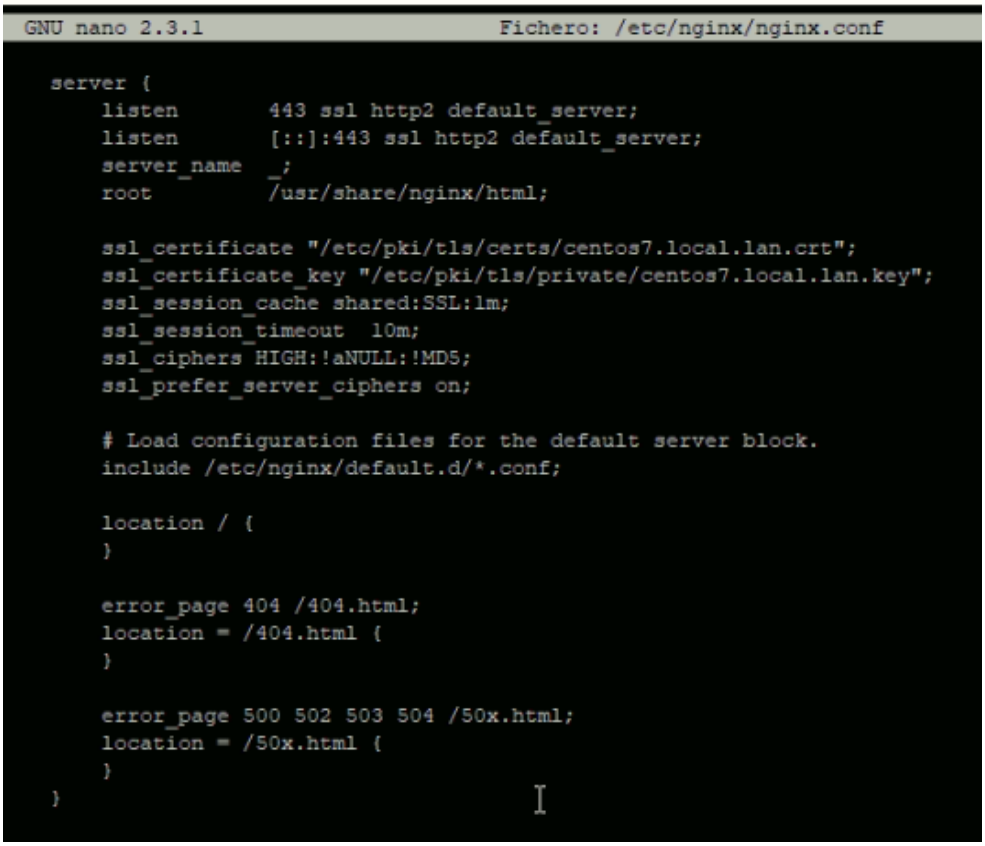
Figura 39. Testear el servicio de *nginx*



Fuente: elaboración propia

Se procede a editar el archivo de configuración de nginx para habilitar el puerto 443 y sus directivas *ssl_certificate* y *ssl_certificate_key* según las rutas especificar las del servidor certificado y su clave privada, como se muestra en la Figura 40.

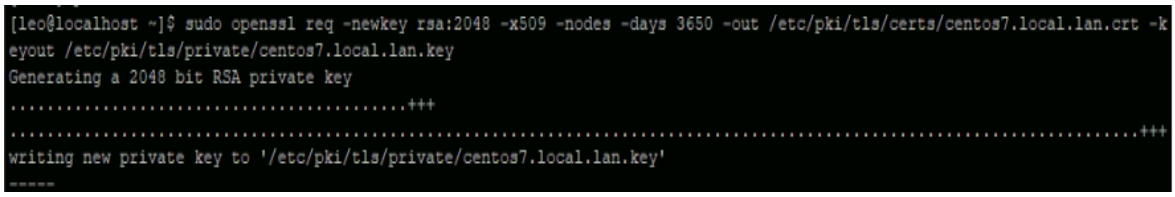
Figura 40. Configuración del archivo de *nginx*



Fuente: elaboración propia

Para este proyecto se usará un certificado auto firmado generado mediante el comando *openssl*, como se muestra en la Figura 41.

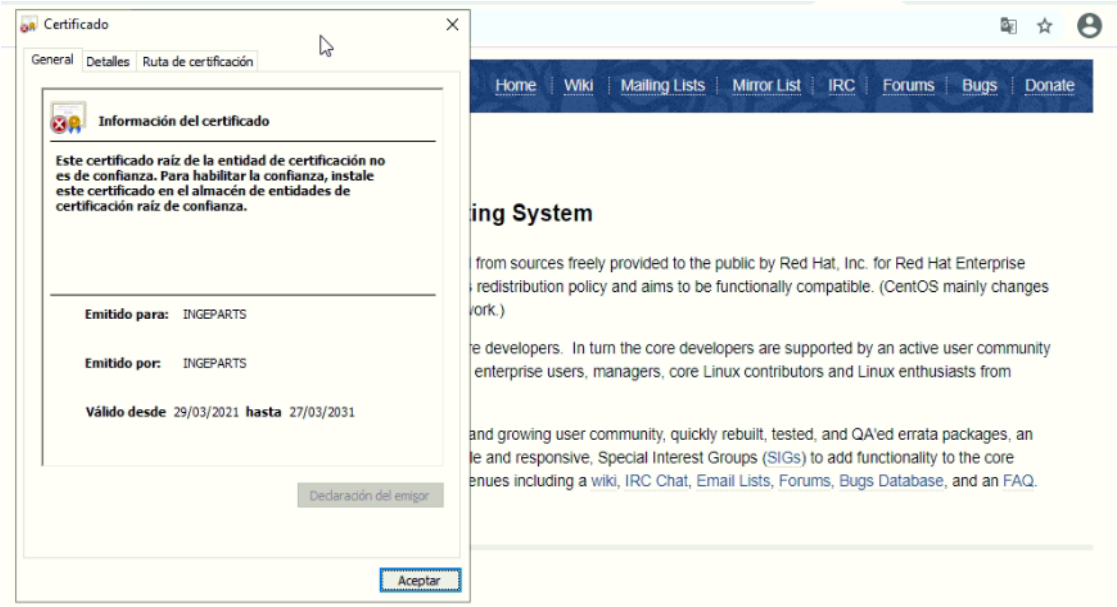
Figura 41. Instalación del certificado de auto firmado



Fuente: elaboración propia

Finalmente, validar y reiniciar el servicio *nginx*, se podrá acceder a la web por defecto tanto con protocolo HTTP como con HTTPS, como se muestra en la Figura 42.

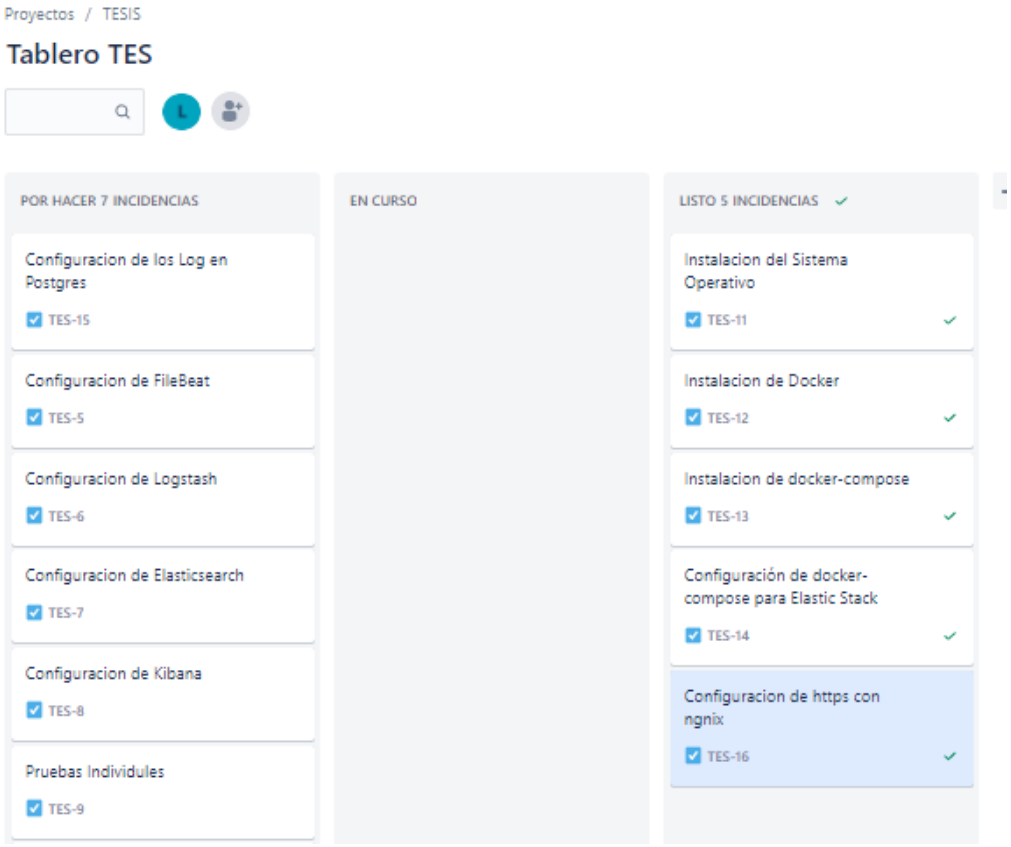
Figura 42. Instalación del certificado de auto firmado



Fuente: elaboración propia

Para finalizar la tarea de la configuración del protocolo *https con nginx*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 43.

Figura 43. Tarea para configuración de https con *nginx*, estado listo



Fuente: elaboración propia

Tarea 6. Configuración de los *logs* en la base de datos *Postgres*

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 44.

Figura 44. Tarea para la configuración de *logs* en postgres, estado en curso

Fuente: elaboración propia

En el presente proyecto se analizará los logs de la base de datos postgres. Para esto se configura algunos parámetros en el archivo postgresql.conf donde está instalado la base de datos postgre como se muestra en la Figura 45.

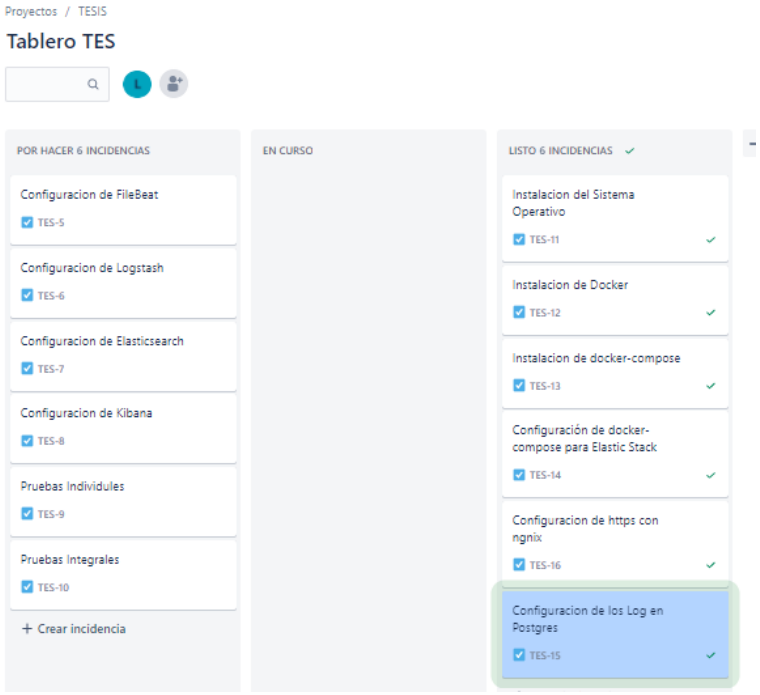
Figura 45. Configuración de *logs* en la base de datos postgres

```
log_checkpoints = on
log_connections = on
log_disconnections = on
log_duration = on
```

Fuente: elaboración propia

Para finalizar la tarea de la configuración de *logs* en *postgres*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 46.

Figura 46. Tarea para configuración de *logs* en postgres, estado listo

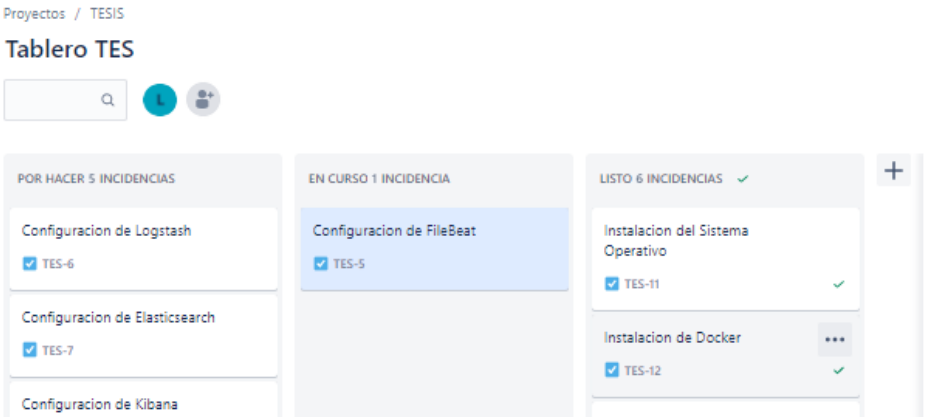


Fuente: elaboración propia

Tarea 7. Configuración de FileBeat

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado en **curso** como se muestra en la Figura 47.

Figura 47. Tarea para la configuración de *filebeat*, estado en curso

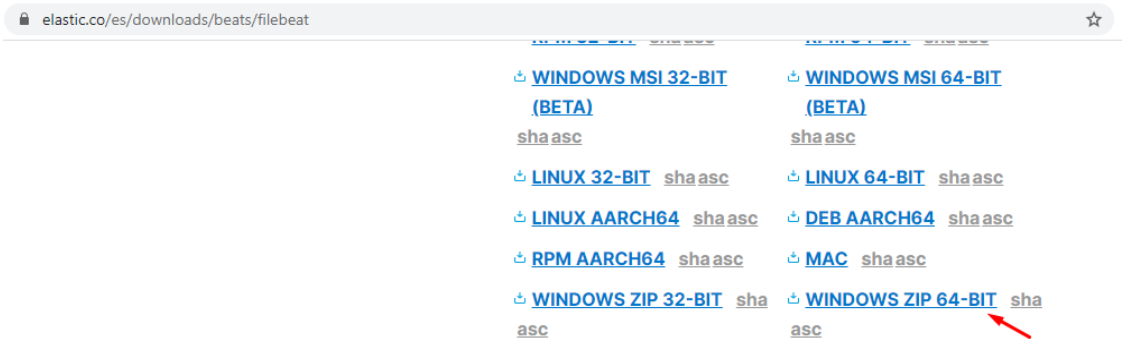


Fuente: elaboración propia

FileBeat se encarga de enviar logs generados en *postgres* a *Logstash*. El gestor de base de datos postgres está instalado en un servidor de Windows, donde se configurará FileBeat con la versión 7.11.2, los pasos para la descarga y configuración de FileBeat son las siguientes:

1. Descargar de la página oficial Elastic el archivo zip, para el sistema operativo Windows de 64 bits como se muestra en la Figura 48.

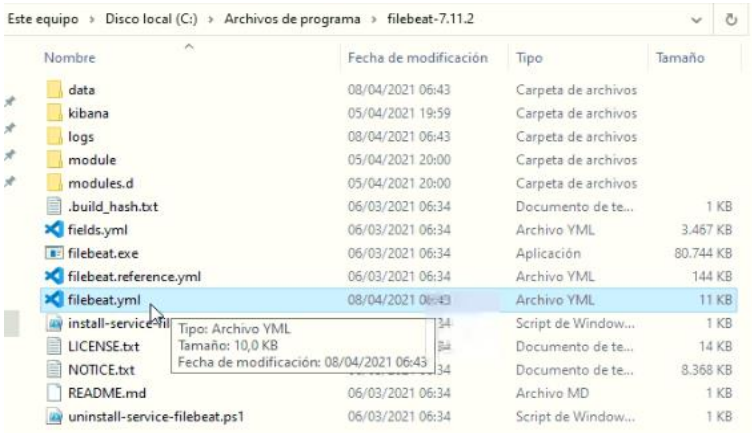
Figura 48. Descargar *FileBeat*



Fuente: elaboración propia

2. Una vez descargado, descomprimir el archivo donde se obtendrá una carpeta con archivos de configuración, esta carpeta poner en una ubicación fácil de ubicar, posterior a esto se identifica el archivo filebeat.yml, como se muestra en la Figura 49.

Figura 49. Identificación del archivo *filebeat.yml*

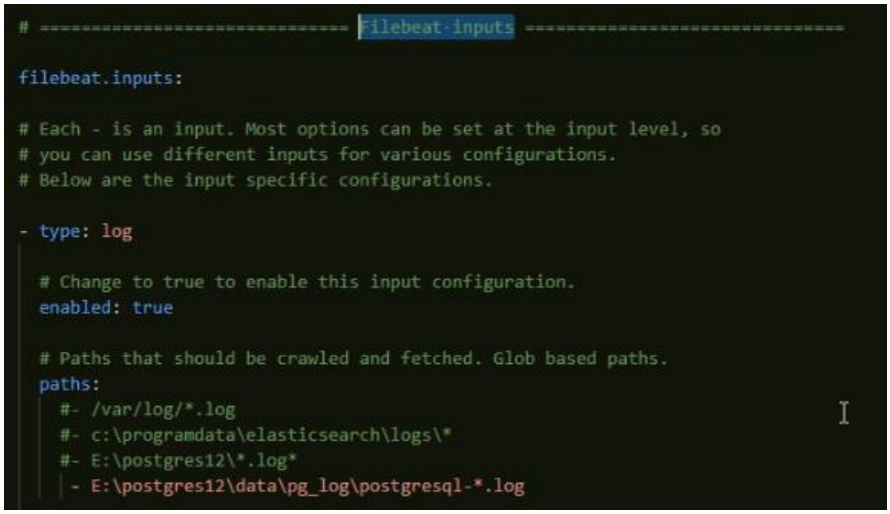


Fuente: elaboración propia

3. Abrir el archivo con VsCode que es un editor de texto avanzado para este tipo de configuraciones, una vez abierto el archivo se realizara dos cambios que son: en el input y el output.

3.1. En el archivo ubicarse en la sección de **Filebeat inputs** donde se cambiará la ubicación de los logs que genera el sistema, como se muestra en la Figura 50.

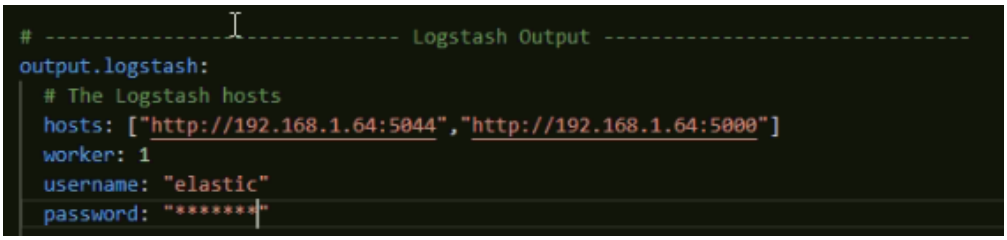
Figura 50. Configuración del input en el archivo *filebeat.yml*



Fuente: elaboración propia

3.2. Como segundo paso ubicarse en la sección **Logstash Output,** donde se va a enviar los logs, como se muestra en la Figura 51.

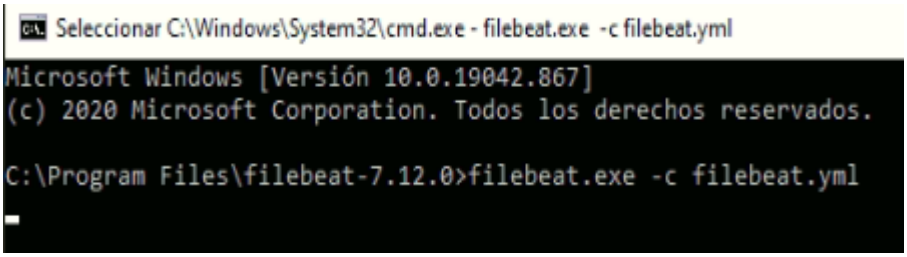
Figura 51. Configuración del output en el archivo *filebeat.yml*



Fuente: elaboración propia

4. Como último paso abrir una consola y ubicarse en la dirección donde está la carpera de filebeat, para enviar a ejecutar el archivo *filebeat.yml*, como se muestra en la Figura 52.

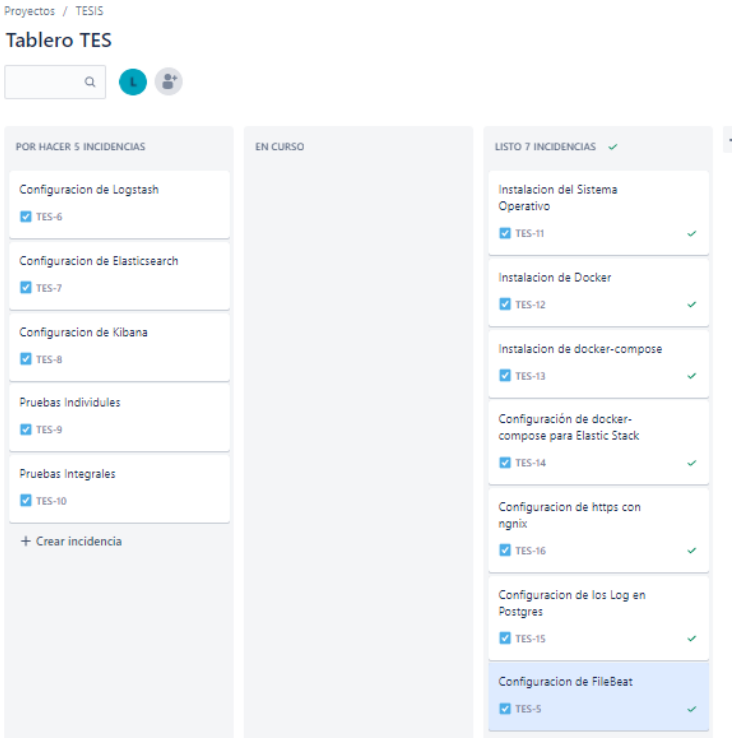
Figura 52. Ejecución del archivo *filebeat.yml*



Fuente: elaboración propia

Para finalizar la tarea sobre la configuración de *filebeat*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 53.

Figura 53. Tarea para configuración de *filebeat*, estado listo

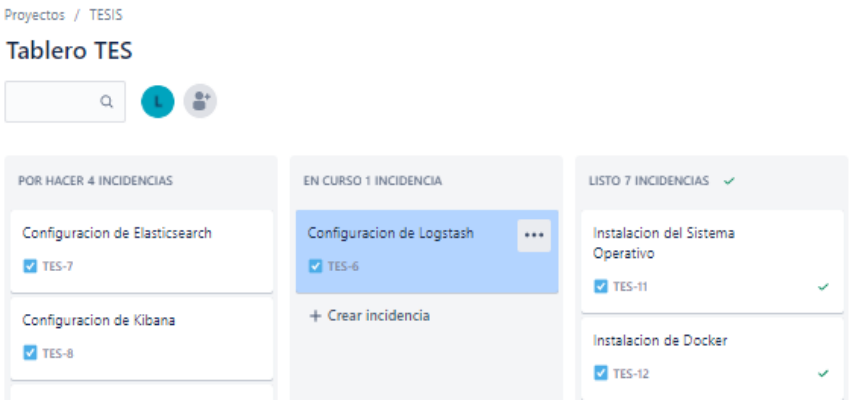


Fuente: elaboración propia

Tarea 8. Configuración de Logstash

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado **en curso** como se muestra en la Figura 54.

Figura 54. Tarea para la configuración de *logstash*, estado en curso

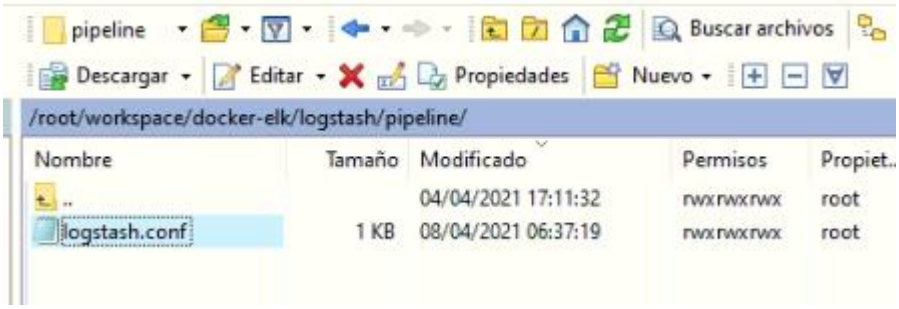


Fuente: elaboración propia

Logstash es el responsable de recibir los logs y procesar la información, para enviar a *elasticsearch* datos limpios, legible y fácil de entender.

En las configuraciones anteriores se instaló *docker-compose* con todas las herramientas del stack de *elastic*, una de estas herramientas es *logstash* donde se va a configurar los parámetros necesarios para su correcto funcionamiento. En el sistema operativo de centos donde se instaló el *stack* de *elastic*, ubicar la carpeta de *docker-elk***,** dentro de esta carpeta dirigirse al siguiente directorio: **logstash/pipeline/,** ahí está el archivo **logstash.conf.** Para abrir este archivo utilizar la herramienta *WinSCP* su principal función es conectarse a una maquina local o en la nube, como se muestra en la Figura 55.

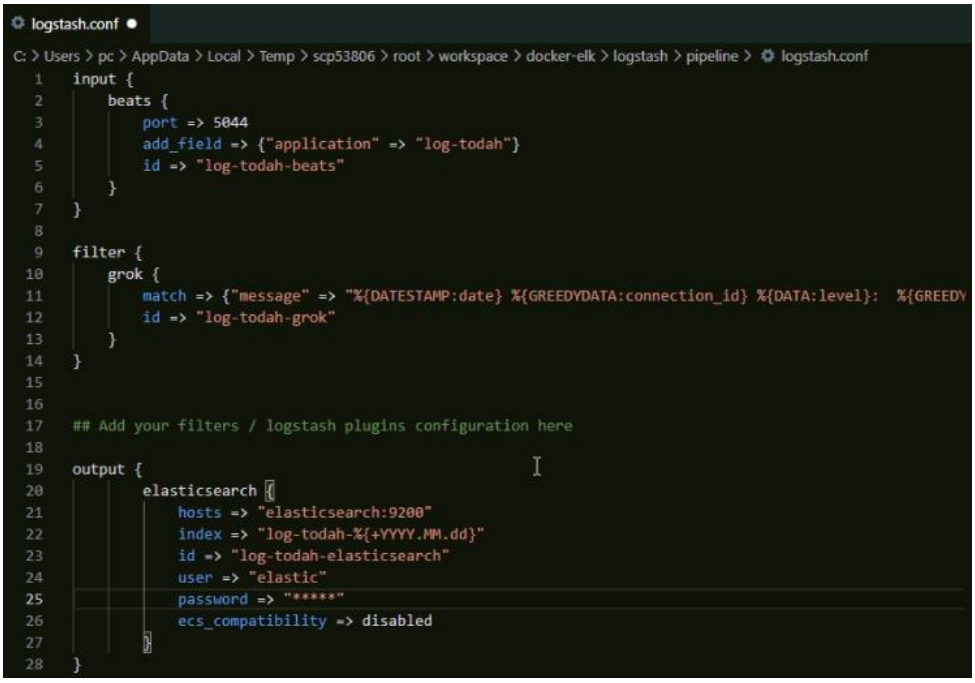
Figura 55. Identificar la ubicación del archivo de configuración de *logstash*



Fuente: elaboración propia

Una vez identificado el archivo, abrir con el editor de texto VsCode para proceder a cambiar los parámetros, para procesar los logs y posteriormente enviar a *elasticsearch*, como se muestra en la Figura 56.

Figura 56. Configuración de *logstash*



Fuente: elaboración propia

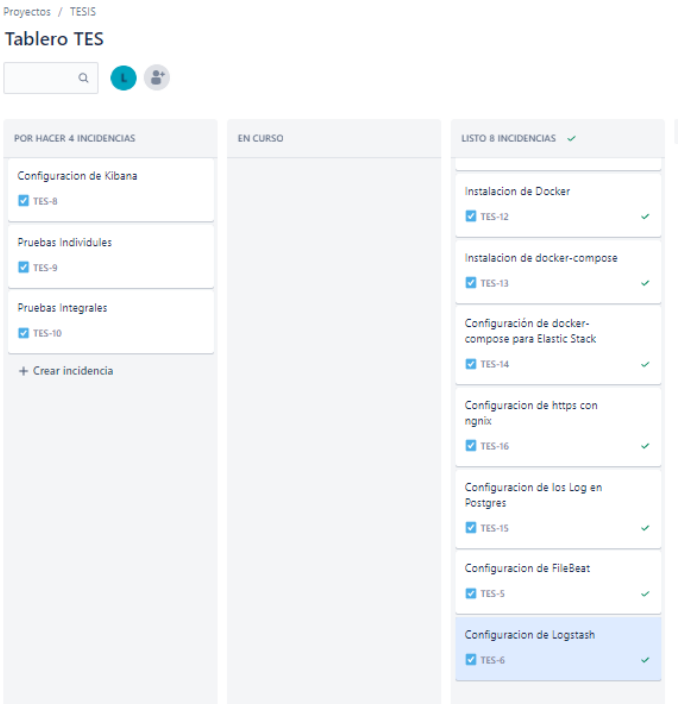
Finalmente, validar que el proceso este correctamente en un navegador, como se muestra en la Figura 57.

Figura 57. Validar el proceso de *logstash*

Fuente: elaboración propia

Para finalizar la tarea sobre la configuración de *logstash*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 58.

Figura 58. Tarea para configuración de *logstash*, estado listo

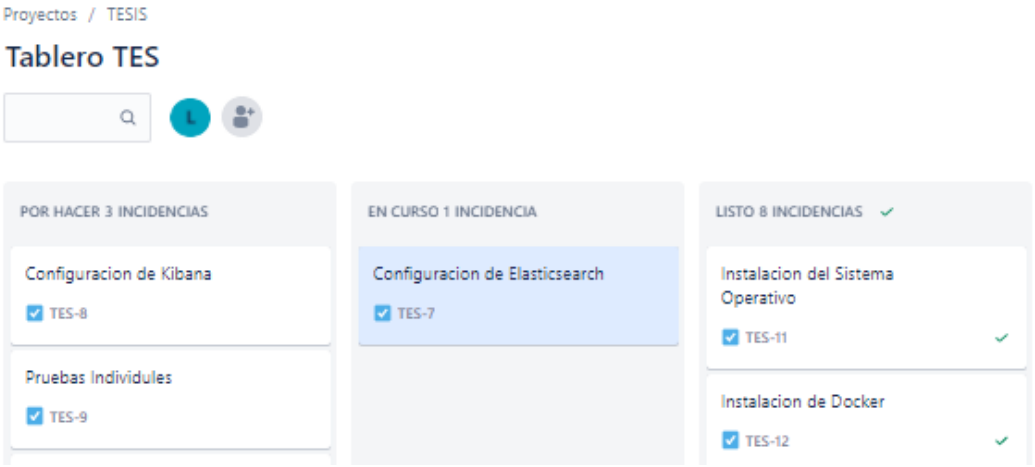


Fuente: elaboración propia

Tarea 9. Configuración de *Elasticsearch*

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado **en curso** como se muestra en la Figura 59.

Figura 59. Tarea para la configuración de *logstash*, estado en curso

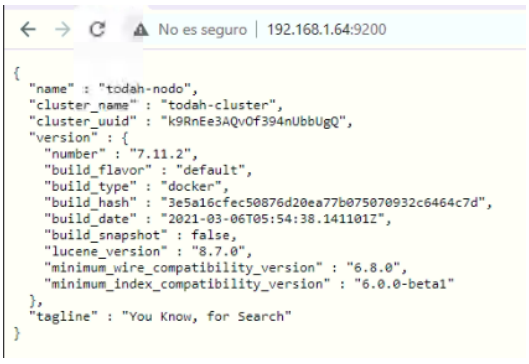


Fuente: elaboración propia

Elasticsearch recibe la información enviada por *logstash* y lo convierte en índices, para procesar la información de forma organizada y asegurar el ciclo de vida de la aplicación.

Para empezar con la configuración validar en un navegador que se ejecute el servicio de *elasticsearch* correctamente, como se muestra en la Figura 60.

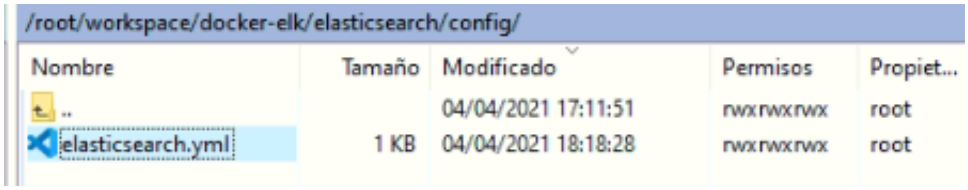
Figura 60. Revisión del servicio de *elasticsearch*



Fuente: elaboración propia

De igual forma como se realizó en la configuración de *logstash* buscar el directorio de *elasticsearch* el archivo elasticsearch.yml, como se muestra en la Figura 61.

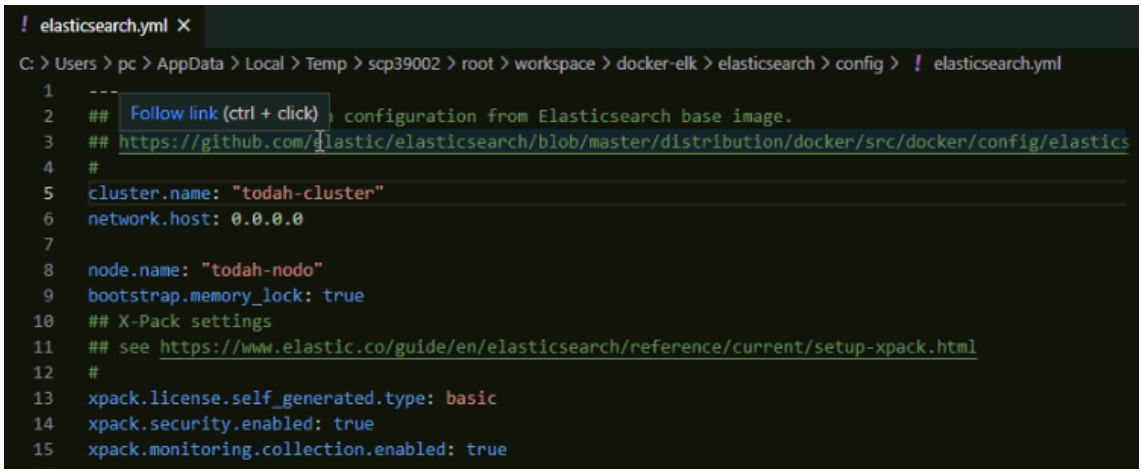
Figura 61. Identificar el directorio de *elasticsearch*



Fuente: elaboración propia

Una vez identificado el archivo, abrir con el editor de texto *VsCode* para proceder a cambiar los parámetros del *cluster* y de los nodos que se ejecutaran para sincronizar la información con *kibana*, como se muestra en la Figura 62.

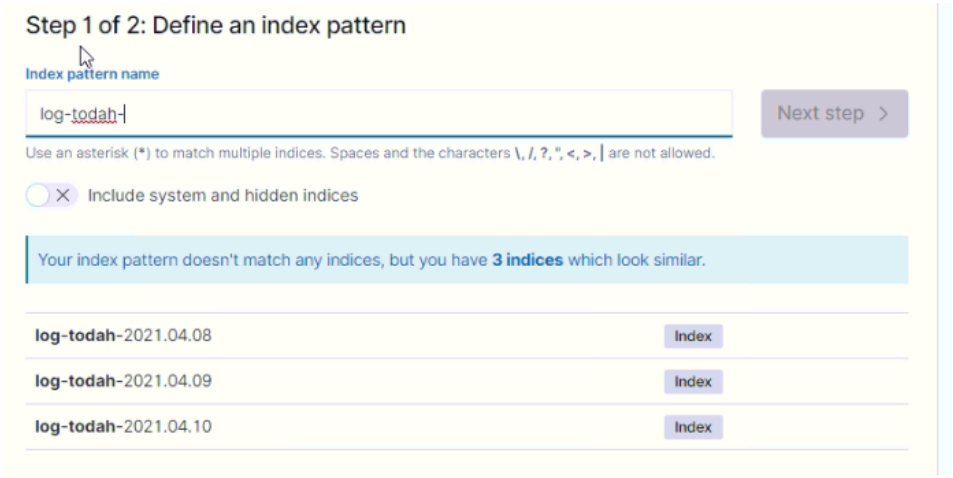
Figura 62. Configuración de *elasticsearch*



Fuente: elaboración propia

Crear los índices que llegan desde *logstash* a *elasticsearch* con el nombre de log-todah-, para poder diferenciar de los demás índices que se crean automáticamente, como se muestra en la Figura 63.

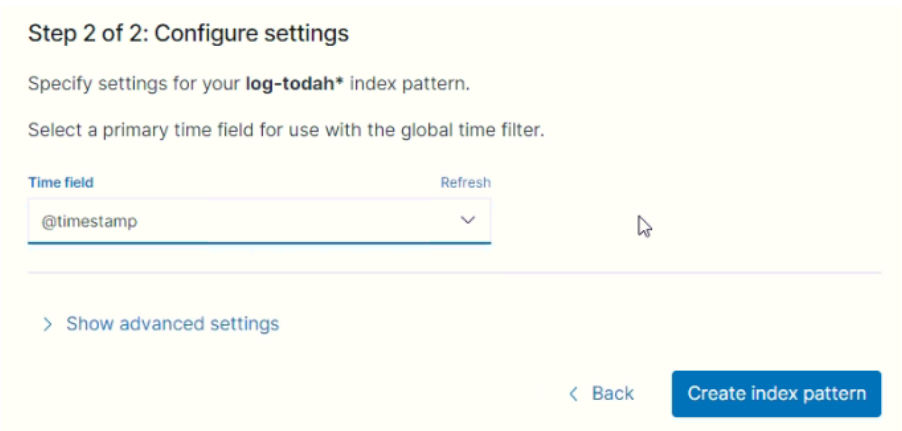
Figura 63. Creación de índices en *elasticsearch* parte1



Fuente: elaboración propia

Como segundo paso seleccionar el campo de tiempo principal para usar con el filtro de tiempo global, como se muestra en la Figura 64.

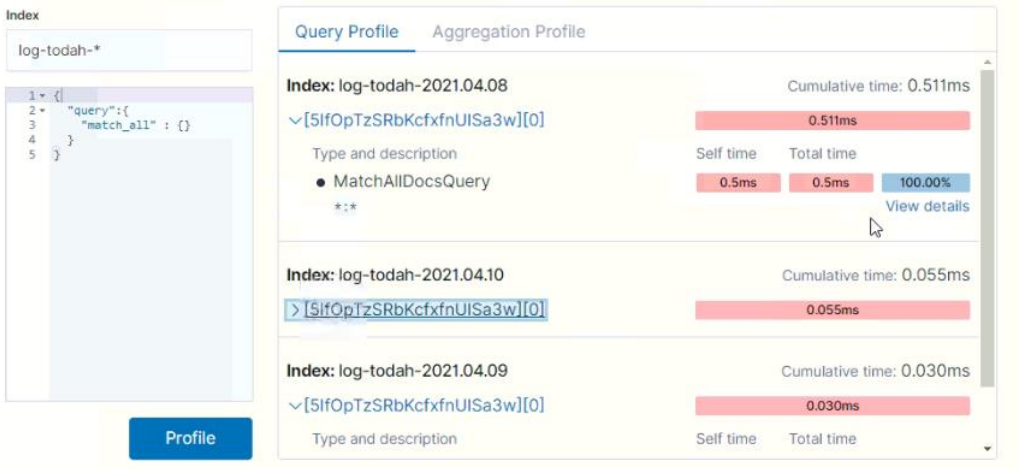
Figura 64. Creación de índices en *elasticsearch* parte2



Fuente: elaboración propia

Para validar que el proceso este correctamente los índices y los tiempos de respuestas sean estables en el tiempo, buscar la opción ***Query Profile*** para ejecutar el siguiente comando, como se muestra en la Figura 65.

Figura 65. Validar los tiempos de respuesta en los índices de *elasticsearch*



Fuente: elaboración propia

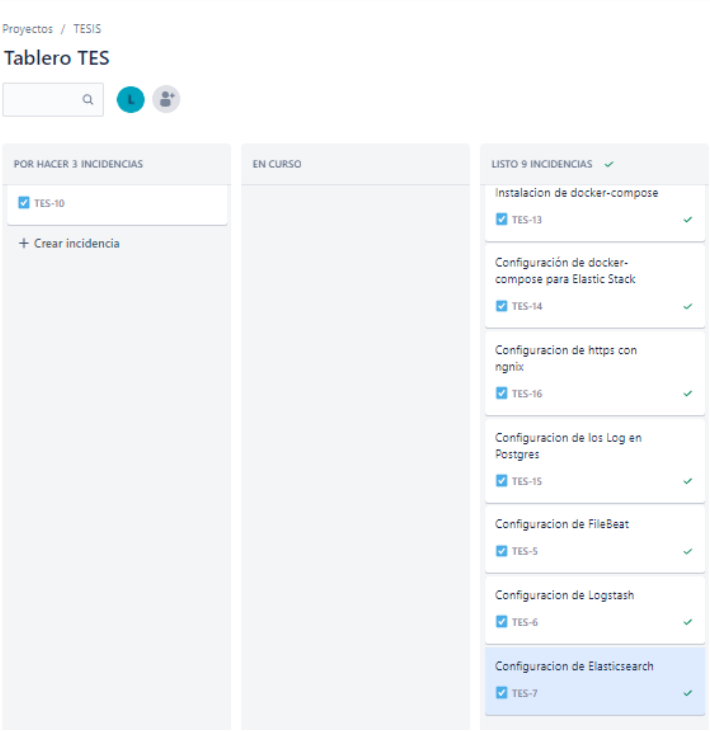
Finalmente, para validar que se haya generado correctamente los índices, en la opción de *Dev Tools* dentro de la consola de *elastichsearch* ejecutar el siguiente comando, como se muestra en la Figura 66.

Figura 66. Validar la creación los índices en *elastichsearch*

Fuente: elaboración propia

Para finalizar la tarea sobre la configuración de *elasticsearch*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 67.

Figura 67. Tarea para configuración de *elasticsearch*, estado listo



Fuente: elaboración propia

Tarea 10. Configuración en Kibana

Antes de iniciar con la configuración, en el tablero de Kanban se mueve la tarea al estado **en curso** como se muestra en la Figura 68.

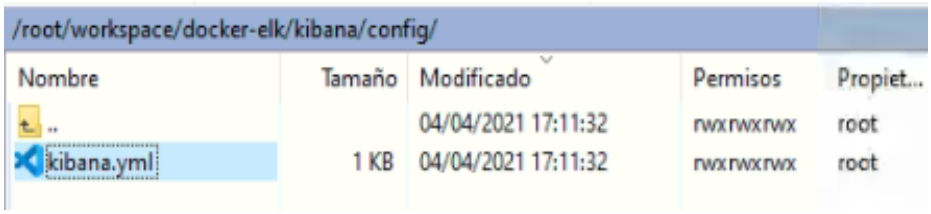
Figura 68. Tarea para la configuración de *kibana*, estado en curso

Fuente: elaboración propia

Kibana ayuda con la visualización los logs de forma dinámica, para toma de decisiones acertadas ante vulnerabilidades, esto asegura el ciclo de vida de la aplicación.

La última herramienta de *elastic stack* es *kibana*, de igual forma como se realizó en las configuraciones de las herramientas anteriores, buscar el directorio de *kibana* el archivo kibana.yml, como se muestra en la Figura 69.

Figura 69. Identificar el directorio de *kibana*



Fuente: elaboración propia

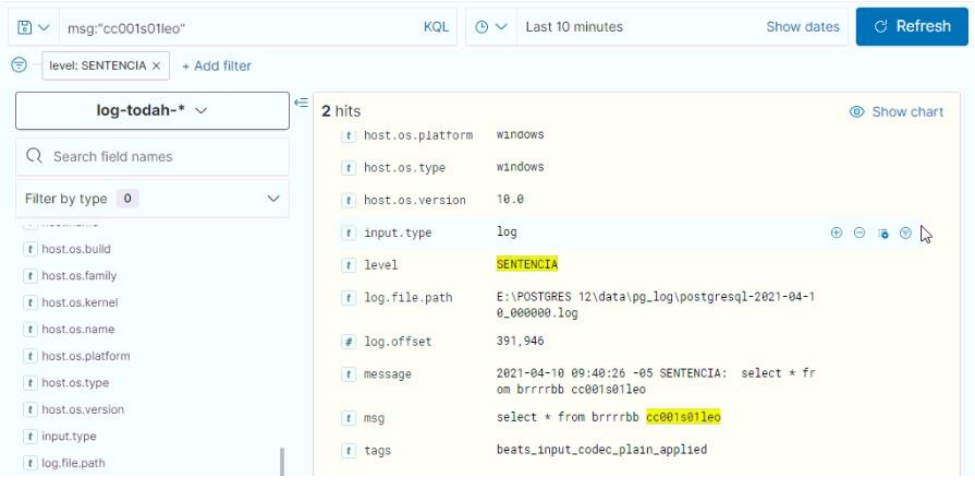
Una vez identificado el archivo, abrir con el editor de texto *VsCode* para proceder a cambiar los parámetros del host donde se visualizará y desplegará, de igual forma las credenciales para que la información llegue de forma segura a kibana, como se muestra en la Figura 70.

Figura 70. Configuración de *kibana*

Fuente: elaboración propia

Para visualizar y filtrar la información de los logs que llega a *kibana*, identificar la opción ***Dicover*** esta opción está dentro de *kibana*, como se muestra en la Figura 71.

Figura 71. Visualización y filtros de *kibana*

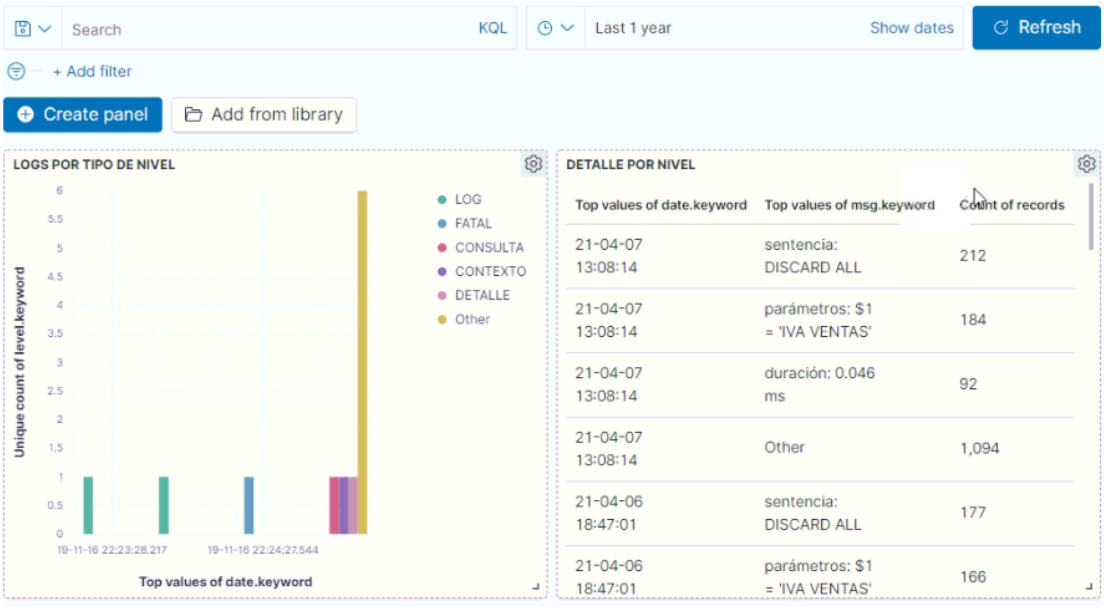


Fuente: elaboración propia

Para una mejor visualización de la información se podría diseñar gráficos dinámicos como es la opción de histogramas.

El primer histograma representa la visualización de las incidencias detallada por nivel y fecha, como se muestra en la Figura 72.

Figura 72. Histograma por nivel



Fuente: elaboración propia

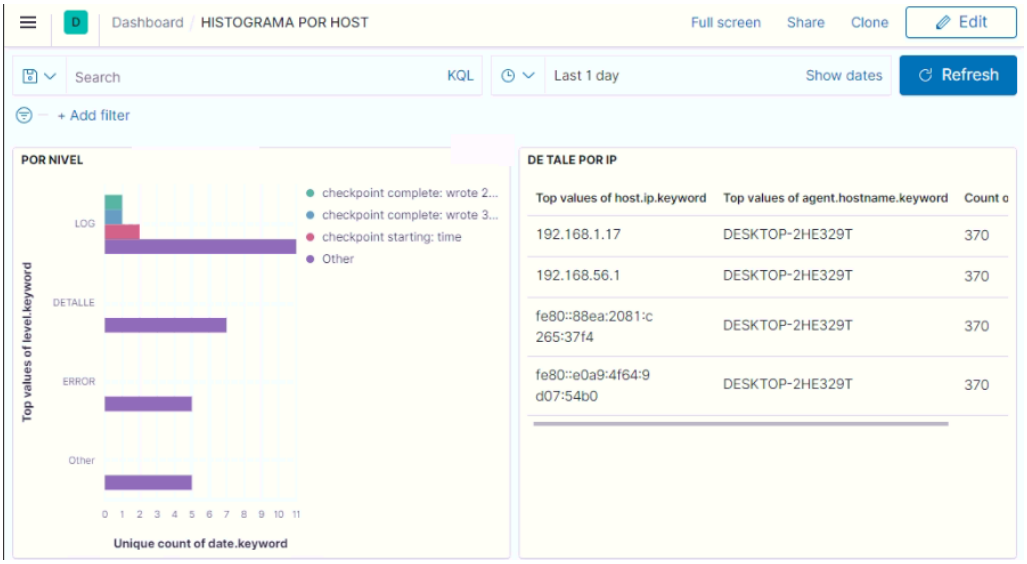
Histograma por origen, visualiza la ruta donde se crearon los logs, representado por un *path* (es la ruta donde está ubicado un archivo), como se muestra en la Figura 73.

Figura 73. Histograma por origen

Fuente: elaboración propia

Histograma por host, se visualiza el origen de la IP y el nombre del computador de donde se envían los logs, como se muestra en la Figura 74.

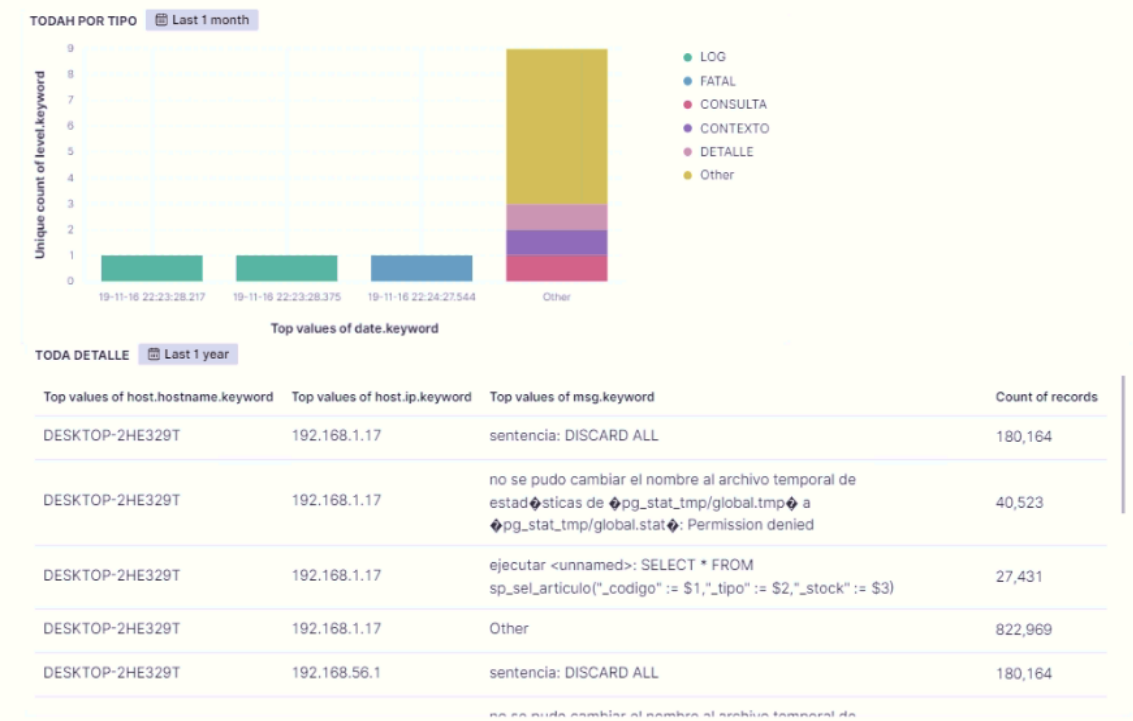
Figura 74. Histograma por *host*



Fuente: elaboración propia

El último paso es configurar la opción *canvas* de *kibana*, su función principal es integrar varias visualizaciones en una sola ventana, como se muestra en la Figura 75. *Canvas* ayuda a mejorar la abstracción de los datos y la toma de decisiones rápidas y eficientes ayuda a mitigar los eventos inadecuado que podría ejecutarse en el sistema.

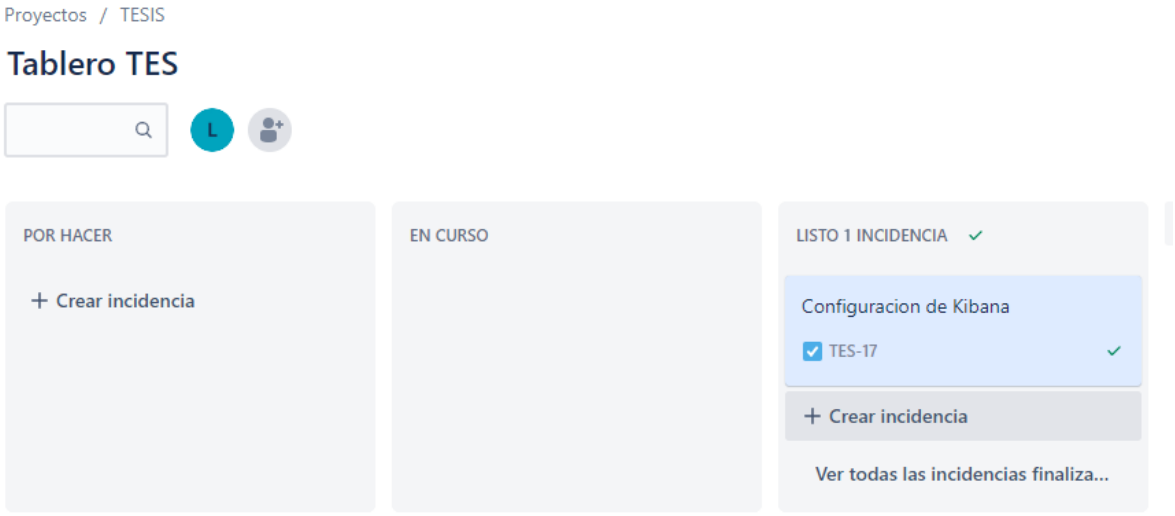
Figura 75. Visualización en tipo *Canvas*



Fuente: elaboración propia

Para finalizar la tarea sobre la configuración de *kibana*, en el tablero de Kanban se mueve la tarea al estado **listo,** como se muestra en la Figura 76.

Figura 76. Tarea para configuración de *kibana*, estado listo



Fuente: elaboración propia

Una vez terminado la implementación de *Elastic Stack*, en el tablero *kanban* todas las incidencias se encuentra en estado listo, es decir, se completó todas las tareas correctamente.
Los *logs* de la aplicación ya se encuentran en monitoreo automatizado, la información se podrá visualizar de forma dinámica en la herramienta de *Kibana*, donde ayuda a la toma de decisiones dentro del ciclo de vida de las aplicaciones con un ambiente escalable según vaya se desarrolle nuevas aplicaciones en la empresa. Los estándares de seguridad implementados fortalecen de forma permanente y automatizada a encontrar vulnerabilidades y poder mitigar para evitar filtrado de información.

Para finalizar la implementación se realiza una descripción de los estándares de seguridad y las etapas del ciclo de vida de la aplicación que se obtiene en cada tarea ejecutada, como se muestra en la Tabla 12.

Tabla 12. Estándares de seguridad y ciclo de vida de la aplicación ejecutada por tarea

Tareas	Estándares de seguridad	Ciclo de vida de la aplicación
Instalación del Sistema Operativo	Se instalo *Centos* 7, es un sistema operativo seguro y es utilizado para desplegar aplicaciones.	Etapa de diseño.
Instalación de *Docker*	Se instalo y configuro *Docker* (contenedores) para tener un ambiente escalable según se implemente aplicaciones en la empresa.	Etapa de diseño.
Instalación de *docker-compose*	Para tener un ambiente seguro y estable que ayude a la configuración de nuevas aplicaciones, se desplego *docker-compose.*	Etapa de diseño.
Configuración de *docker-compose* para Elastic Stack	El despliegue y configuración de *Elastic Stack* se clono del repositorio oficial, para asegurar la integridad, disponibilidad y confidencialidad de la información de los logs generados por el sistema.	Etapa de implementación.
Configuración de https con *ngnix*	Para asegurar la información se implementó un certificado de seguridad.	Etapa de implementación.
Configuración de los *Log* en Postgres	En el ciclo de vida de la aplicación se encuentra un componente importante que es la base de datos, donde se configura los logs de forma agrupada que genera el sistema.	Etapa de implementación.
Configuración de *FileBeat*	Los logs que genera el sistema se envían de forma encriptada con el método sha256 al *Logstash*.	Etapa de implementación.
Configuración de *Logstash*	Para tener la información limpia y depurada con filtros personalizados se utilizó *Logstash*.	Etapa de implementación.
Configuración de *Elasticsearch*	El almacenamiento de todos los logs recopilados se almacena en *Elasticsearch*, categorizados por índices para una mejor visualización y seguridad de la información.	Etapa de implementación.
Configuración de *Kibana*	La visualización de forma dinámica se realiza con la herramienta de *Kibana* que ayuda a tener una visualización en tiempo para realizar tomas de decisiones ante posibles ataques.	Etapa de pruebas, documentación y mantenimiento.

Fuente: elaboración propia

CAPÍTULO III. ANÁLISIS DE LOS RESULTADOS DE LA INVESTIGACIÓN

3.1 Análisis de resultados

Los análisis de definiciones relacionados con el proceso del monitoreo en el ciclo de vida de las aplicaciones, es una necesidad en la actualidad, porque permiten una mejor visualización de lo que se monitorea en tiempo real para una toma de decisiones correcta y efectiva ayuda, también, a corregir las vulnerabilidades que existen en las aplicaciones. Es decir, el proceso de monitoreo es importante y necesario para tener aplicaciones seguras y confiables.

En el estado inicial el monitoreo de aplicaciones en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., no se evidenciaba estándares de seguridad para el monitoreo en el ciclo de vida de las aplicaciones, lo cual, provocaba tiempos tardíos en la revisión manual de las posibles vulnerabilidades en las aplicaciones, por este motivo existió la necesidad de implementar estándares de seguridad con el objetivo de tener la información segura y confiable enfocada a la confiabilidad, integridad y disponibilidad en las aplicaciones de la empresa.

Un aporte a nivel local es tener centralizado el monitoreo de las aplicaciones para asegurar la información, analizar y mitigar las vulnerabilidades que existen en ellas.

Resumen de las pruebas

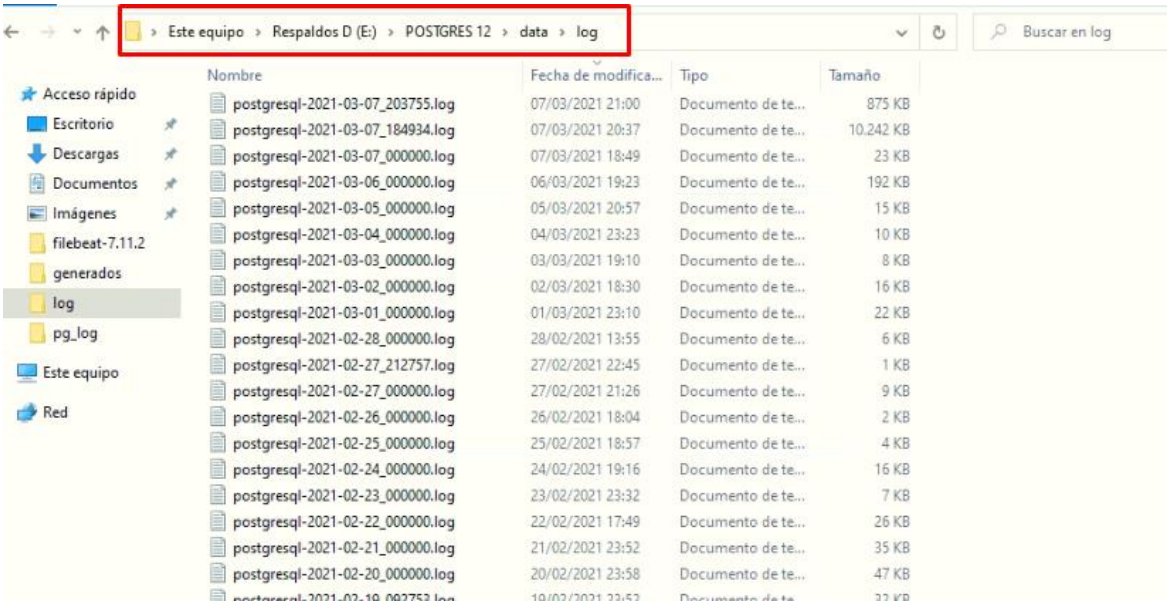
En este capítulo, se realiza dos tipos de escenarios: escenario 1 (como estaba antes de la implementación) y escenario 2 (como quedo después de la implementación), estos escenarios son aplicados con los estándares de seguridad en la aplicación TODAH ERP en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda. Esta aplicación es un ERP (Planificación de Recursos Empresariales) con los siguientes módulos: Compras, Ventas, Financiero, Caja y Bancos, Anexo Transaccional, Reportes Avanzados, configuraciones e Inteligencia de negocios, toda la información esta alojada en la base de datos *Postgres*. Fue creado en base al ERP Microsoft Dynamics GP en el año 2016.

3.1.1 Escenario 1

Antes de implementar los estándares de seguridad en el ciclo de vida de las aplicaciones, el monitoreo se realizaba manualmente, como se muestra en los siguientes pasos:

Cada fin de mes o si el sistema estaba lento se procedía a revisar los logs en la siguiente dirección, como se muestra en el Gráfico 5.

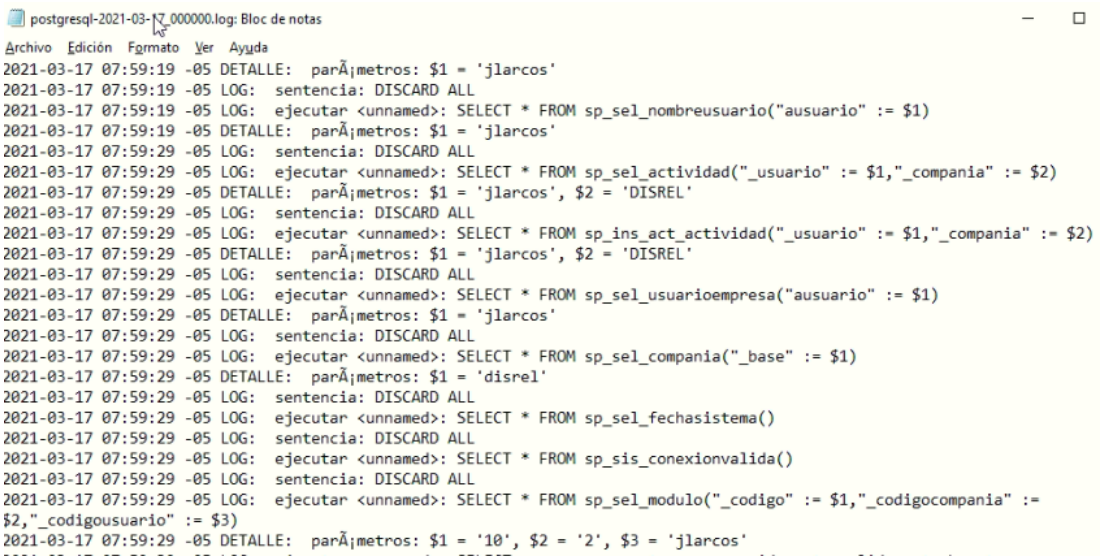
Gráfico 5. Escenario 1, dirección de los logs



Fuente: elaboración propia

Una vez localizado los archivos generados por el sistema se procedía abrir con un editor de textos, como se muestra en el Gráfico 6.

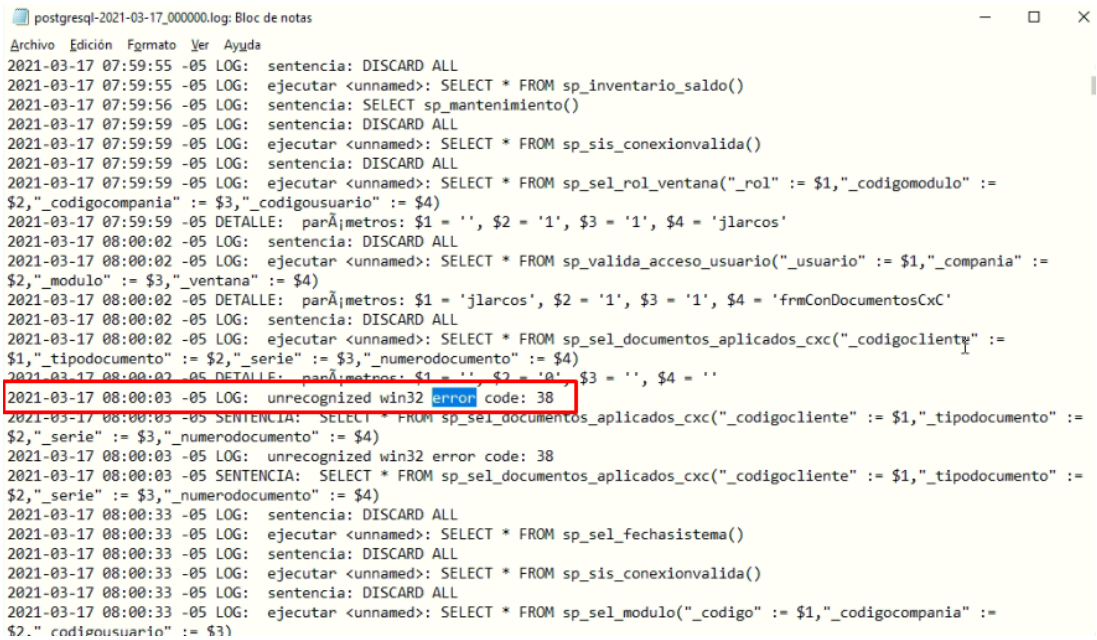
Gráfico 6. Escenario 1, visualización de los logs



Fuente: elaboración propia

Con el archivo del log abierto, se procedía a buscar errores o alertas en el funcionamiento del sistema, como se muestra en el Gráfico 7.

Gráfico 7. Escenario 1, identificación de errores o alertas



Fuente: elaboración propia

El proceso era tedioso, tocaba revisar todas las líneas para poder identificar si existe alguna inconsistencia en el funcionamiento del sistema y poder corregir. Todo este proceso generaba tiempo y costo a la empresa y solo se realizaba cada fin de mes o si los usuarios reportaban que el sistema estaba lento o las transacciones generaban datos erróneos.

A continuación, se presenta en la Tabla 13, en la que, se detalla las características de los estándares de seguridad antes de la implementación.

Tabla 13. Escenario 1, Resumen de resultados

CARACTERÍSTICA	CUMPLE	NO CUMPLE
Informes dinámicos		X
Despliegue en la nube con *elastic enterprise*		X
Escalabilidad en las aplicaciones		X
Datos limpios		X
Monitoreo de las aplicaciones *	X	

Fuente: elaboración propia

*** Monitoreo de las aplicaciones**

Como se visualiza en el escenario 1 se muestra el proceso de monitoreo de las aplicaciones, pero solo enfocado a la funcionalidad como se muestra en el Grafico 7.

3.1.2 Escenario 2

Después de implementar los estándares de seguridad en el ciclo de vida de las aplicaciones, el monitoreo se hizo dinámico y fácil de entender sobre los logs generados por el sistema.

Para la visualización del monitoreo en el ciclo de vida de las aplicaciones se realizó las pruebas con 5 usuarios, para que ingrese al aplicativo TODAH ERP simultáneamente con el objetivo de revisar los logs de alerta, advertencia y errores que se podrán generar al momento de realizar transacciones o consultas en el sistema.

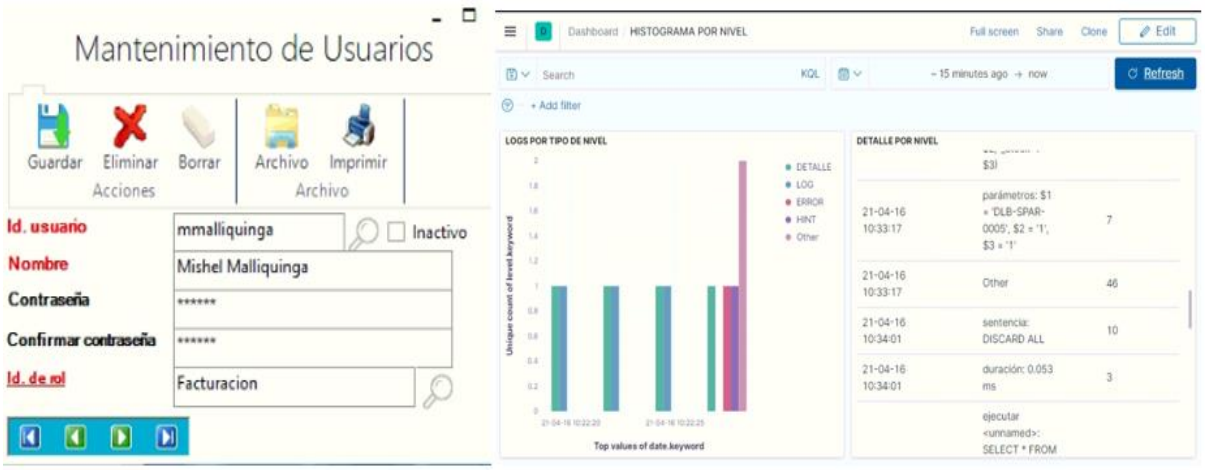
Los logs generados se podrán visualizar gráfica y detalladamente al momento que los usuarios intentaron ingresar al sistema con credenciales incorrectas, como se muestra en el Gráfico 8.

Gráfico 8. Escenario 2, ingreso al sistema TODAH ERP

Fuente: elaboración propia

De igual forma se procedió a cambiar las credenciales de los usuarios dentro del sistema. Esta información generada, también, se podrá visualizar en un gráfico dinámico en *kibana*, como se muestra en el Gráfico 9.

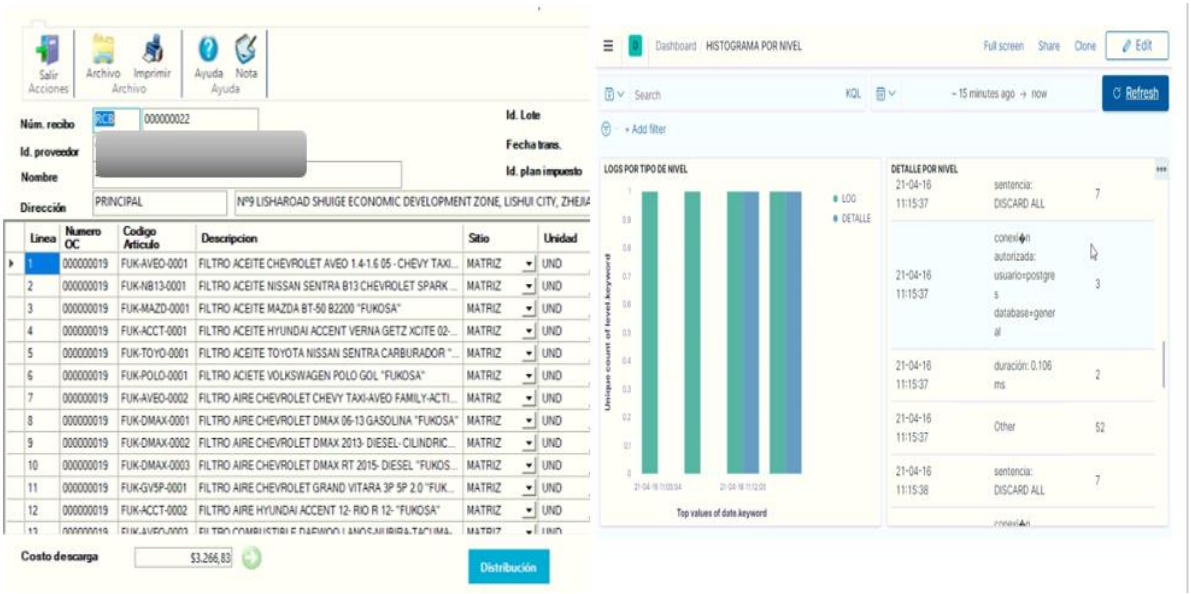
Gráfico 9. Escenario 2, cambio de credenciales en el sistema TODAH ERP



Fuente: elaboración propia

Para validar los tiempos de respuesta en una transacción en el sistema, se procedió a ingresar una importación, visualizado el tiempo de respuesta que fue de 0.106 ms (milisegundos), como se muestra en el Grafico 10.

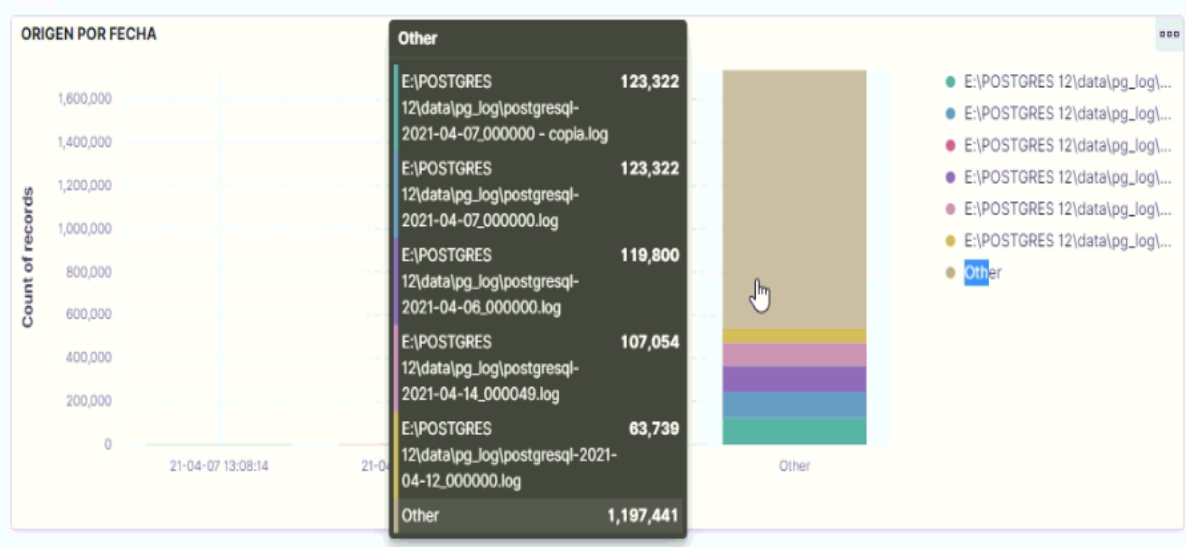
Gráfico 10. Escenario 2, ingreso de transacción en el sistema TODAH ERP



Fuente: elaboración propia

Finalmente, se podrá evidenciar la dirección donde se almacenan los logs de la aplicación categorizada por fecha, como se muestra en el Gráfico 11.

Gráfico 11. Escenario 2, dirección de los logs en el sistema TODAH ERP



Fuente: elaboración propia

A continuación, se presenta en la Tabla 14, en la que, se detalla las características de los estándares de seguridad después de la implementación.

Tabla 14. Escenario 2, Resumen de resultados

CARACTERÍSTICA	CUMPLE	NO CUMPLE
Informes dinámicos *	X	
Despliegue en la nube con *elastic enterprise*		X
Escalabilidad en las aplicaciones **	X	
Datos limpios ***	X	
Monitoreo de aplicaciones ****	X	

Fuente: elaboración propia

*** Informes dinámicos**

En el escenario 2, ya se realiza un monitoreo de las aplicaciones mediante informes dinámicos como se muestra en el Grafico 11.

**** Escalabilidad en las aplicaciones**

Por medio de la herramienta *docker*, las aplicaciones se podrán acoplar en un ambiente escalable.

***** Datos limpios**

Para la obtener datos limpios de los logs, se utilizó la herramienta *Logstash*, que ayuda depurar los datos como se muestra en la Figura 26.

****** Monitoreo de aplicaciones**

Con la implementación de los estándares de seguridad en el ciclo de vida de las aplicaciones se monitorea en tiempo real los sucesos que se generan en el sistema, todo esto enfocado a la funcionalidad y seguridad de la información, como se evidencia en la Grafico 8.

CONCLUSIONES

- La fundamentación teórica acerca de los estándares de seguridad para el aseguramiento del ciclo de vida de las aplicaciones evidencia que las aplicaciones son desarrollas para su funcionamiento y la seguridad queda en un segundo plano. Para ello el S*tack* de *Elastic* brinda los estándares de seguridad para el ciclo de vida de las aplicaciones.

- El diagnostico en la filtración de información y vulnerabilidades de la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda, pone en evidencia, cuáles son las amenazas en el ciclo de vida de la aplicación, esto produce que el departamento de sistemas tome decisiones rápidas y efectivas con estándares de seguridad en un ambiente escalable.

- Los estándares de seguridad implementada en las aplicaciones con el S*tack* de *Elastic,* se evidencia las vulnerabilidades existentes en todo el procedo del ciclo de vida de la aplicación, razón por, la cual, se mitiga y corrige rápidamente, para tener una aplicación que asegura la información de la empresa.

- La verificación del funcionamiento del S*tack* de *Elastic,* con la herramienta *Kibana*, se monitorea la aplicación en tiempo real con informes dinámicos, para una mejor visualización de las posibles vulnerabilidades en el ciclo de la aplicación, permite priorizar las amenazas criticas según el tipo de *logs* que genere la herramienta.

RECOMENDACIONES

- Se recomienda a todas las empresas privadas y públicas, tener estándares de seguridad para el monitorio en los procesos del ciclo de vida de las aplicaciones, a fin de mitigar las vulnerabilidades, se podrá generar en el código fuente, base de datos o ataques a las aplicaciones por usuarios no autorizados.

- A la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda., se recomienda monitorear constantemente la aplicación con la herramienta *Kibana,* a fin de detectar amenazas o errores en el aplicativo para poder notificar al proveedor del sistema y proceda a corregir el problema con el objetivo de tener una aplicación segura y confiable.

- Se recomienda utilizar *Elastic Stack Enterprise* en la nube, con el objetivo de tener actualizado todas las herramientas de monitoreo de *Elastic*, en una estructura redúndate, escalable y con tolerancia a fallos para asegurar la continuidad del negocio y tener aplicaciones seguras.

- La tecnología actualmente es cambiante, por lo que se recomienda tener el S*tack* de *Elastic*, alojado en un repositorio en la nube con integración continua, para automatizar procesos de creación, pruebas e implementación con el objetivo de tener aplicaciones seguras enfocadas a la Confidencialidad, Integridad y Disponibilidad de la información.

BIBLIOGRAFÍA

Andrés, R. (2017, mayo 31). *Qué es una máquina virtual, cómo funciona y para qué sirve*. ComputerHoy. https://computerhoy.com/noticias/software/que-es-maquina-virtual-como-funciona-que-sirve-46606

Ataques Web. (2018, enero 1). https://revista.seguridad.unam.mx/numero-05/ataques-web

Atlassian. (2019a, enero 1). *¿Para qué sirve Jira?* Atlassian. https://www.atlassian.com/es/software/jira/guides/use-cases/what-is-jira-used-for

Atlassian. (2019b, marzo 1). *Kanban frente a scrum*. Atlassian. https://www.atlassian.com/es/agile/kanban/kanban-vs-scrum

AWS. (2020, enero 1). *Contenedores de Docker | ¿Qué es Docker? | AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/docker/

Canalnews. (2020, mayo 29). Vulnerabilidades en aplicaciones móviles. *Canal News Ecuador*. https://canalnewsecuador.com/2020/05/29/vulnerabilidades-en-aplicaciones-moviles/

Carrera, D. (2013, septiembre 23). AnalisisDiseño: CICLO DE VIDA DE UNA APLICACIÓN INFORMÁTICA. *AnalisisDiseño*. http://informaticosbustamantecevallos.blogspot.com/2013/09/ciclo-de-vida-de-una-aplicacion.html

CEC. (2020, mayo 20). *ISO/IEC 27034: Estándar Internacional para la seguridad de las aplicaciones | noticias.cec.es*. https://noticias.cec.es/index.php/2020/03/20/isoiec-27034-estandar-internacional-para-la-seguridad-de-las-aplicaciones/

Centos. (2018, octubre 8). *¿Qué es CentOs? ¿Que significa y que hace? ¿Es Linux? ¿RedHat? ¿Open Source? | Blog HostDime Latinoamérica, servidores dedicados*. https://www.hostdime.la/blog/que-es-centos-que-significa-y-que-hace-es-linux-redhat-open-source/

Coelho, F. (2019, mayo 17). *Significado de Metodología*. Significados. https://www.significados.com/metodologia/

Conislla, F. (2020, febrero 14). *Seguridad en el ciclo de vida del desarrollo de software*. Belatrix Software Development Blog. https://www.belatrixsf.com/blog/seguridad-desarrollo-software

Cuervo, P. V. (2019, febrero 20). *Contenedores versus Máquinas Virtuales*. Arquitecto IT. http://www.arquitectoit.com/docker/contenedores-vs-maquinas-virtuales/

Docker-Compose, D. (2021, marzo 12). *Overview of Docker Compose*. Docker Documentation. https://docs.docker.com/compose/

García, P. V. (2019, abril 22). ELK Stack: Kibana, Elasticsearch y Logstasg - Monitorización | Enimbos. *Enimbos.com*. https://enimbos.com/monitorizacion-de-aplicaciones-usando-elk-stack/

Google Maps. (2019, enero 1). *INGEPARTS*. 1°15’44.9"S 78°34’04.7"W. https://www.google.com/maps/place/1%C2%B015'44.9%22S+78%C2%B034'04.7%22W/@-1.262459,-78.5701666,796m/data=!3m2!1e3!4b1!4m5!3m4!1s0x0:0x0!8m2!3d-1.262459!4d-78.5679779?hl=es

Guévin, M. (2018, septiembre 6). El ciclo de vida de un programa y la gestión del ciclo de vida de las aplicaciones (ALM) en Agile. *Nutcache*. https://www.nutcache.com/es/blog/el-ciclo-de-vida-de-un-programa-y-la-gestion-del-ciclo-de-vida-de-las-aplicaciones-alm-en-agile/

Henriquez, P. (2020, abril 29). COVID-19: ¿Una oportunidad para la transformación digital de las pymes? *Puntos sobre la i*. https://blogs.iadb.org/innovacion/es/covid-19-oportunidad-transformacion-digital-pymes/

Hostinger. (2017, septiembre 7). ¿Qué Es El Protocolo SSH Y Cómo Funciona? *Tutoriales Hostinger*. https://www.hostinger.co/tutoriales/que-es-ssh

ISW. (2015, noviembre 25). Ventajas KANBAN. *Kanban* . https://iswkanban.wordpress.com/ventajas-y-desventajas/

Kanbanize. (2018, enero 4). *Qué es Kanban: Definición, Características y Ventajas*. Kanban Software for Agile Project Management. https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban

López, P. por F. (2018, noviembre 7). Contenedores Docker | Blog Conasa. *Servicios y soluciones IT Conasa*. https://www.conasa.es/blog/contenedores-docker/

Magic Online. (2021, marzo 3). *¿Por qué alquilar un servidor en la nube empresarial?* Magic Online. https://www.magiconline.es/cloud/por-que-alquilar-un-servidor-en-la-nube-empresarial/

Manger, E. (2020, febrero 1). *¿Qué es el monitoreo del rendimiento de aplicaciones? | Software de monitoreo de aplicaciones—ManageEngine Applications Manager*. https://www.manageengine.com/latam/applications_manager/que-es-el-monitoreo-de-aplicaciones.html

OKDIARIO. (2019, enero 27). *Observación directa: Un método para recolectar datos*. okdiario.com. https://okdiario.com/curiosidades/conoce-metodo-observacion-directa-3628568

OnLine, P. C. (2020, mayo 11). *La tríada CIA: Definición, componentes y ejemplos | CambioDigital OnLine*. https://cambiodigital-ol.com/2020/03/la-triada-cia-definicion-componentes-y-ejemplos/

OWASP-LATAMTour-Patagonia-2016-rvfigueroa.pdf. (s. f.). Recuperado 21 de noviembre de 2020, de https://owasp.org/www-pdf-archive/OWASP-LATAMTour-Patagonia-2016-rvfigueroa.pdf

Palma, A. E. L., Hurtado, X. G. B., Ron, J. L., Mozo, P. J. M., Montoya, D. R. D., & Quiñónez, D. F. B. (2019). *La observación. Primer eslabón del método clínico*. 9.

PowerData, G. (2020, enero 2). *Seguridad de datos: En qué consiste y qué es importante en tu empresa*. https://www.powerdata.es/seguridad-de-datos

Red Hat. (2020, enero 1). *¿Qué es la gestión del ciclo de vida de las aplicaciones (ALM)?* https://www.redhat.com/es/topics/devops/what-is-application-lifecycle-management-alm

Regina. (2018, octubre 10). ¿Qué es JIRA y para qué sirve? *Agilmania*. https://agilmania.com/que-es-jira/

ReYDeS. (2018, mayo 11). *Proyecto OWASP para un Estándar de Verificación en la Seguridad de Aplicaciones | Alonso Caballero / ReYDeS*. http://www.reydes.com/d/?q=Proyecto_OWASP_para_un_Estandar_de_Verificacion_en_la_Seguridad_de_Aplicaciones

Ricote, S. I. (2019). *Seguridad en el ciclo de vida del desarrollo del software: DevSecOps*. 74.

Rodriguez, C. G. (2020, abril 18). Guía básica para aplicar la técnica de la entrevista en investigación. *Tesis de Cero a 100*. https://tesisdeceroa100.com/guia-basica-para-aplicar-la-tecnica-de-la-entrevista-en-investigacion/

Rouse, M. (2019, enero 1). *¿Qué es Docker? - Definición en WhatIs.com*. SearchDataCenter en Español. https://searchdatacenter.techtarget.com/es/definicion/Docker

Serrotho. (2020, septiembre 18). *Orquestación de Contenedores*. https://blog.techdata.com/ts/latam/orquestación-de-contenedores

SGSI. (2015, mayo 21). *ISO 27001: Seguridad de la Información*. https://www.pmg-ssi.com/2015/05/iso-27001-que-significa-la-seguridad-de-la-informacion/

SIMPLYEASYLEARNING. (2020, enero 1). *Software—Ciclo de Vida de Desarrollo—Tutorialspoint*. https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm

Sosa, A. (2013, octubre 11). *El Método Analítico-sintético*. prezi.com. https://prezi.com/c3cu3jwuax79/el-metodo-analitico-sintetico/

Toledo Garcia, R. (2017, febrero 10). Diferentes metodologías ágiles |. *DIFERENTES METODOLOGÍAS ÁGILES*. https://lorbada.com/blog/2017/02/10/diferentes-metodologias-agiles/

WMWARE. (2020, abril 2). *Aplicaciones seguras*. https://www.vmware.com/topics/glossary/content/application-security.html

ANEXOS

Anexo 1: Ficha de observación directa

FICHA DE OBSERVACIÓN DIRECTA	
Ficha N.:	01
Investigador:	Fredy Leonardo Arroba Flores
Motivo para observar:	Obtener información del monitoreo en el ciclo de vida de las aplicaciones.
Observaciones: La presente observación es para constatar el monitoreo en el ciclo de vida de las aplicaciones implementadas o que están en desarrollo en la empresa Ingeparts Importaciones Automotrices PG&QE Cia.Ltda, actualmente se realiza manualmente el monitoreo, esto consume tiempo laboral en el departamento de sistemas.	

Anexo 2: Formato de la entrevista

Entrevista

El objetivo de la entrevista es conocer cómo se ejecuta el monitoreo en el ciclo de vida de las aplicaciones enfocado a la seguridad de la información.

Las respuestas que proporcione a las preguntas serán absolutamente confidenciales, para la recolección y análisis de datos de este estudio.

1. ¿Cuán importante es realizar el monitoreo de las aplicaciones?

   _____

2. ¿Cada que tiempo se dedica a realizar el monitoreo de las aplicaciones?

   _____

3. ¿Tiene estándares de seguridad para el monitoreo de las aplicaciones?

   _____

4. ¿Cómo realiza el monitoreo de los logs en las aplicaciones?

   _____

5. ¿Aplica alguna herramienta para monitoreo de las aplicaciones?

_____

6. ¿Realiza monitoreo en ciclo de vida de las aplicaciones?

_____