FACULTAD 3

Título: Módulo Monitoreo de logs del sistema Digilante.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor: Ailyn Zulueta González

Tutores: Ing. Rafael Corpas Crehuet

Ing.Osiel Sánchez Martínez

La Habana, septiembre 2020

"Año 61 de la Revolución"

Declaración Jurada de Autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a 8 días del mes de septiembre del año 2020.

Ailyn Zulueta González

_____

Firma del Autor

Ing. Rafael Corpas Crehuet

_____

Firma del Tutor

Ing. Osiel Sánchez Martínez

_____

Firma del Tutor

Agradecimientos

Agradezco a mis padres por apoyarme en todo momento y porque además hoy sehacen realidad sus sueños.

A mi mamá por ser la mejor madre del mundo, por su apoyo, su cariño y su amor, por todo lo que me ha brindado en la vida,por confiar en mi a pesar de las dificultades, por desear este título másque yo, por susconsejos cuando en algún momento intenté abandonar, gracias a ti hoy estoy aquí.

A mi papá por sentirse siempre orgulloso,por ver en mí y en mis hermanos sus sueños hechos realidad, por su amor de padre, por su mano amiga. A él, quiero decirle donde quiera que esté que la niña de sus ojos es ya ingeniera en ciencias informáticas.

A mis hermanos por estar pendientes a mí todo este tiempo, por todo su apoyo y cariño brindado.

A mi novio por estar y esperarme.

A mis tutores Osiel y Rafael, por la preocupación, ocupación, entrega y paciencia en la realización de este trabajo, sin ustedes no lo hubiese logrado. A Osiel que más que un tutor ha sido un amigo, te mereces el premio al mejor tutor, a Rafael por poner a mi disposición todo el conocimiento que posee, por guiarme y hacerme más fácil el camino hacia este momento. A ambos gracias por ser mis guías en este trayecto, por estar conmigo en los momentos más importantes.

Al tribunal por todas las sugerencias brindadas, han sido de mucha ayuda yenseñanza.

A todos mis profesores a lo largo de la carreraque han contribuido a mi formación como profesional, en especial a lasprofesoras(es) Karenia, Hilda, Juan Gabriel, Claudia, Yordanis, Josué, Yanet, Pedro, Yadian Betancourt.

A mis amigos y amigas, las de antes, las de ahora, las de toda la vida, por ser mi paño de lágrimas, misconsejeras, mis defensoras, por no tener la obligación de estar y haber estado, por ser simplemente, misamigas. A mi piquete querido de la uci Rosángela, Karla y Milena por darme la oportunidad de compartir a sulado momentos que nunca olvidaré y que espero que no se queden ahí, a mis amigasDarlyn y Yaimi Olivia por no solo compartir la adrenalina de los Juegos Mellas sinotambién ser compañeras y amigas, además de tener siempre una taza de café para mí, a mi amigo Julio Luis por arrastrarme a todos los eventos la FEU y hacerme sentir de corazón universidad. A mis compañeros de aula por compartir siempre la mejor Fiesta de los Países.

A todos los que de una forma u otra ha aportado a este triunfoen mi vida y espero que estemos siempre juntos.

Dedicatoria

Dedico esta tesis a mi mamá Gleisy, por todo su sacrificio dedicado a mí durante estos años.

A mi papá Luis Ernesto, que donde quiera que este sé que está muy orgulloso de mí, porque siemprelo estuvo desde día que ingrese en esta universidad.

A mis hermanos Luis Ernesto y Kevinpor todo el apoyo que me handado.

A Lázaro por acompañarme en todo momento.

A mis tutores Osiel y Rafael por todo su apoyo paciencia y sabiduría.

A todos los que hacen suyos mis triunfos.

"El trabajo será creador al máximo y el mundo deberá estar interesado en su trabajo y en el de los demás, en el avance de la sociedad día a día"

Ernesto "Che" Guevara de la Serna

Resumen

El sistema de videovigilanciaDigilante, implementado en la empresa de tecnologías de la información para la defensa (XETID), está diseñado para responder a los desafíos de seguridad de los entornos actuales, ofreciendo flexibilidad, escalabilidad y el control necesarios para una administración integral de la seguridad física. Presenta una arquitectura orientada a microservicios, donde cada servicio genera una serie de registros (logs) relacionados con el funcionamiento del sistema, de los cuales se necesita su monitoreo para un posterior trabajo con los datos. En la presente investigación se desarrolla el módulo Monitoreo de Logs para el diagnóstico y corrección de fallas del sistema. Se analizan los elementos teóricos que fundamentan la investigación, así como la metodología de desarrollo, las herramientas y tecnologías utilizadas durante la implementación del módulo en la propuesta de solución. Se representó la arquitectura definida, así como, los requisitos funcionales y no funcionales que rigieron el proceso de desarrollo del mismo.

Palabras Claves: archivos de registros, cámara de vídeo,logs, recolección, seguridad, sistemas de videovigilancia, soporte digital.

Abstract

The Digilante video surveillance system, implemented in the defense information technology company (XETID), is designed to respond to the security challenges of today's environments, offering the flexibility, scalability and control necessary for comprehensive management of physical security. It presents an architecture oriented to microservices, where each service generates a series of logs related to the operation of the system, of which its analysis and recovery is needed for a later work with the data. In this investigation, the Logs Monitoring module is developed for the diagnosis and correction of system failures. The theoretical elements underlying the research, as well as the development methodology, tools and technologies used during the implementation of the module for the solution proposal are analyzed. The architecture defined for the module was represented, as well as the functional and non-functional requirements that governed its development process.

Keywords: log files, video camera, logs, collection, security, video surveillance systems, digital support.

Índice

Índice de Ilustraciones

Índice de Tabla

Introducción

Las Tecnologías de la Información y Comunicaciones (TIC), han permitido llevar el desarrollo tecnológico a todas las esferas de la comunicación, facilitando la conexión entre las personas e instituciones a nivel mundial. Su difusión se refleja en actividades como las redes sociales, las bases masivas de información, medios electrónicos, arte digital y administración de actividades comerciales. Estas incluyen la electrónica como tecnología base que soporta el desarrollo de las telecomunicaciones, la informática y el audiovisual.(1)

El desarrollo de las TIC ha logrado un gran avance en todos los campos de la sociedad incluyendo la videovigilancia, como es el caso de los Sistemas de Gestión de Cámaras de Videos, compuestos por varios subsistemas interrelacionados, para lograr un todo, implementando una arquitectura de microservicios.(2)

Existen sistemas de videovigilancia que proporcionan a las entidades las herramientas adecuadas para proteger a las personas, asegurar los activos y garantizar el cumplimiento regulatorio de seguridad. Creados para integrar múltiples aplicaciones y dispositivos de seguridad (vídeo cámaras, controladores de acceso, sensores, sistemas automatizados de detección de intrusos y sistemas automatizados contra intruso), permiten aprovechar el hardware ya instalado a medida que se amplía el sistema. Las opciones modulares de hardware y software hacen que sea fácil y económico expandir el sistema para mantenerse a la par de las crecientes demandas del negocio. (2)

Cuba no se encuentra aislada a este creciente avance tecnológico, tal es el caso de la Empresa de Tecnologías de la Información para la Defensa (XETID), la cual se encuentra desarrollando el sistema de videovigilanciaDigilante para responder a los desafíos de seguridad de los entornos actuales del país. Dicha aplicación tiene la misión de ofrecer la flexibilidad, escalabilidad y el control necesario para una administración integral de la seguridad física. Posee una arquitectura orientada a microservicios, donde cada uno de estos está distribuido y generan una serie de registros (logs) relacionados con el funcionamiento correcto o no del sistema. El monitoreo de estos registros son de vital importancia a la hora de realizar un diagnóstico sobre el funcionamiento íntegro del sistema; así como para efectuar una posterior corrección de fallas.

Los registros no son solamente archivos en los que se van registrando los errores. En estos debe contenerse toda la información que pueda ser relevante para estudiar la evolución del sistema, pudiendo tener un buen número de archivos diferentes, además de información de seguimiento registrada en las bases de datos.(2)

Sin embargo, actualmente en el sistema Digilante el usuario no es capaz de manipular esta información por la ausencia del procesamiento de los datos que aporten a la futura toma de decisiones de la empresa.

A partir de la problémica existente surge el **problema a resolver**: ¿cómo centralizar los registros generados por los distintos servidores del sistema Digilante para facilitar el monitoreo de los mismos?; tomando en cuenta el problema antes propuesto se define como **objeto de estudio** los procesos de centralización y la monitorización de registros. La investigación presenta como **objetivo general:**desarrollar un módulo para el sistema Digilante que permita el monitoreo y centralización de registros generados por los diferentes servidores, identificándose como **campo de acción:** los procesos de centralización y la monitorización de registros en sistemas de videovigilancia.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Realizar un análisis del estado del arte para el desarrollo del marco teórico alrededor del objeto de estudio.
2. Elaborar los artefactos propuestos por la metodología seleccionada para tener un acercamiento de las necesidades a desarrollar.
3. Implementar el módulo Monitoreo de Logs del sistema de videovigilanciaDigilante.
4. Validar la solución propuesta a partir de pruebas.

Con la culminación de estos objetivos se esperan obtener el siguiente **resultado**:
1. Un módulo integrado al sistema Digilante que permita: el monitoreo de registros de distintos servidores, la visualización de los resultados de los registros a través de gráficas y tablas, y la exportación de los mismos en formato pdf.

Métodos científicos

El Método Científico está constituido por un conjunto de pasos o etapas bien establecidas que posibilitan dirigir el proceso de investigación de forma óptima, de modo que permita alcanzar su propósito, el conocimiento científico, de la manera más eficiente. (3)

Método científico es la forma organizada, sistemática y sistémica de estudiar el mundo circundante para llegar al conocimiento y comprensión de los objetos, fenómenos y procesos que lo constituyen. (3)

Existen dos tipos de métodos científicos de la investigación: los métodos de investigación teóricos y los métodos de investigación empíricos.

Para el desarrollo de esta investigación y el logro de su objetivo, se utilizan ambos métodos, a través de los cuales se obtiene una idea más clara y concisa de lo que se pretende realizar.

Métodos teóricos:

Permiten descubrir en el objeto de investigación las relaciones esenciales y las cualidades fundamentales, no detectables de manera sensoperceptual. Por ello se apoya básicamente en los procesos de abstracción, análisis, síntesis, inducción y deducción. (3)

Entre los diferentes tipos de métodos teóricos se emplearon:

Analítico-Sintético: Mediante una profunda investigación para la fundamentación teórica del tema a investigar, se sintetizó toda la información que se consideró importante para el desarrollo del trabajo, aprovechándose del estudio de conceptos, aristas, tecnologías y herramientas adecuadas para la monitorización de registros en sistemas de video vigilancia.

Histórico-Lógico: Fue necesario un análisis del tema a investigar para un estudio detallado de las características que especifican el estado del arte de la problemática, aprovechándose las ventajas y desventajas de cada una de las herramientas necesarias y la investigación de procesos de monitorización de registros en diferentes sistemas para obtenerse así la confección del módulo.

Modelación: Se emplea para generar abstracciones entendibles en vista de explicar la realidad en la investigación, es el patrón a seguir mediante un modelo que se crea con el Lenguaje Modelado Unificado (UML) para representar los procesos que requiere un módulo monitorizado de logs y a la vez modelar la funcionalidad del generador de reportes.

Métodos empíricos

Su aporte al proceso de investigación es resultado fundamentalmente de la experiencia. Estos métodos posibilitan revelar las relaciones esenciales y las características fundamentales del objeto de estudio, accesibles a la detección sensoperceptual, a través de procedimientos prácticos con el objeto y diversos medios de estudio. (3)

Existen diferentes tipos de métodos empíricos. En el caso de nuestra investigación se utilizó:

Entrevista: Se emplea en la fase inicial para el entendimiento de la problemática existente en el sistema de videovigilanciaDigilante. Se realiza una entrevista híbrida, ya que se utilizó la entrevista abierta y la entrevista cerrada, puesto que se realizaron preguntas previamente elaboradas y otras que surgieron en el momento, con el objetivo de realizar un levantamiento de los requisitos funcionales y no funcionales a partir de la información acerca de las características que debía tener dicho sistema.

Observación: Utilizada para el mejor entendimiento del sistema de video vigilancia Digilante, observando las funcionalidades que se despliegan en este producto, tanto en ejecución como en su etapa de implementación; facilitando definir las funcionalidades que estructuraran al módulo Monitoreo de Logs. Utilizando de este método técnicas tales como: observación participante y la observación sistemática.

Estructura de los Capítulos:

Capítulo 1:Fundamentación teórica:

Abarca el estudio del arte a partir de la fundamentación teórica del proceso para monitorizar registros en sistemas similares a Digilante existentes en el contexto actual. Especifica conceptos importantes, características fundamentales, análisis de las herramientas, metodologías, lenguajes de programación y tecnologías que serán usadas durante el proceso de desarrollo del módulo.

Capítulo 2:Fundamentación de la propuesta de solución:

Analiza de forma detallada las tecnologías, herramientas y metodología a utilizar en la solución al problema planteado. Define requisitos funcionales y no funcionales, presenta un modelo propuesto de solución donde se refleja los elementos y aspectos necesarios para la posterior implementación del módulo.

Capítulo 3:Validación de la propuesta de solución:

Se detalla el proceso de implementación del módulo propuesto según el modelo determinado en el Capítulo 2. Finalizada la implementación se realizan pruebas unitarias y pruebas de aceptación con el cliente, a la herramienta obtenida, describiéndose los principales artefactos generados, tales como las actas de liberación interna de productos de software y de aceptación del cliente. Además, se describen los resultados de la aplicación de las métricas para validar el diseño y las técnicas para la validación de los requisitos.

Capítulo 1: Fundamentación teórica

1.1. Introducción

En este capítulo se realiza un estudio conceptual acerca de los términos relacionados con el proceso de videovigilancia y de los sistemas de monitoreo de registros asociados a dicho proceso existentes en el mundo, se describen datos fundamentales como características, funcionalidades, herramientas, tecnologías, metodologías, entre otros. Se describe el objeto de estudio para un mejor entendimiento de la problemática existente, teniendo en cuenta la importancia y necesidad de la solución para el sistema Digilante.

1.2. Conceptos Asociados al dominio del problema

Para un mejor entendimiento del tema a investigar se explican las siguientes definiciones técnicas de los términos mencionados durante la etapa de investigación:

Los **sistemas de video vigilancia** no son más que soluciones para la vigilancia global en el cual se asume como objetivo principal la protección de personas y activos en todos los ambientes (aeropuertos, ferrocarriles, carreteras, puentes, puertos, áreas residenciales, comerciales) y ante todas las causas posibles: sabotajes, terrorismo, robo o vandalismo en general. Utilizándose la tecnología digital como patrón descriptivo de sucesos o hechos que ocurran dentro del local o institución a proteger; mediante el uso de la captura y gestión de videos monitoreados en tiempo real. Dichos sistemas incluyen en su interconexión equipos electrónicos, como cámaras, computadoras, cables de red, dispositivos de interconexión que permiten el flujo constante de datos que provienen de los diferentes eventos. El control automatizado que despliegan los sistemas de video vigilancia propicia también en los entornos monitorizados la inteligencia artificial, que se define en analizar e interpretar por sí solo la incidencia de elementos que requieran alertar al personal encargado de la vigilancia. Mediante los productos diseñados para la actividad de video vigilancia se modifica el patrón de utilizar personal para la tarea de vigilar recursos, que pueden comportarse como humanos y materiales. Las aplicaciones usadas con el fin de cumplir con la actividad de vigilancia para administrar sucesos que surgen durante su despliegue hacen uso de logs. (4)

Los **archivos de registros (Logs)** son los archivos de texto generados por un sistema determinado en respuesta a un evento, registran documentación, comportamiento y condiciones relevantes para un sistema en particular, organizados por relación y colocados al alcance para su uso en cualquier momento tanto por el usuario como por el software para poder administrar toda la información durante un período de tiempo determinado, permitiendo así la solución de problemas y evitando que vuelvan a ocurrir. Se producen automáticamente con fecha y hora.

Los logs son almacenados en los sistemas para su administración, mediante un recurso de cómputo y memoria llamado **ficheros** que es una unidad lógica de almacenamiento que crea el sistema operativo y cuyo contenido especifica el usuario que trata con esta unidad. Los ficheros

están constituidos por un conjunto estructurado y jerárquico de registros lógicos; que almacenan un conjunto de datos virtualmente y pueden ser accedidos y modificados por medio de una computadora. Estos ficheros se identifican por la extensión y el nombre que le asigne el usuario; la extensión es en dependencia de la información y la clasificación del programa que requiera guardar dichos datos (texto, audio o video). Se administra de forma automática y se estructura mediante carpetas y subcarpetas que son localizadas en el disco duro. (4)

Se consideran **reportes** los eventos que agrupan información de acuerdo con un interés específico, donde se muestran resultados de una búsqueda determinada, los cuales pueden ser presentados en forma de textos o a través de gráficos; con los datos reportados se pueden realizar funcionalidades, como cálculos, agrupamientos, patrones estadísticos y mostrarlos al usuario que desea conocerlos. Los reportes son generados dinámicamente, porque cada vez que se llaman o se invocan se actualizan los datos que contienen; la función de estos es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios (5). Son eventos continuos que convergen en resultados deseados; utilizando gráficas, figuras, letras o símbolos que en el mundo se definen como medio de comunicación.

Una **Interfaz de Programación de Aplicaciones (API)** es la que proporciona un conjunto de funciones y procedimientos que puedan ser usados por otro software, dándoles un uso general.

De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, y evitando tener que programar todo desde el principio.

Resumiéndolo, la API sería pre-programar un sistema que consiste en una serie de funciones, las cuales ejecutan cosas en tu aplicación.

Es una forma de dar un acceso restringido a la información de tu aplicación sin dejar que el programador externo acceda a tu código. Solo puede usar y ejecutar aquellas funciones que tú le has dejado preparadas con el permiso correspondiente. (6)

1.3. Logs

¿Qué es un log? Se conoce como log o historial de logs (registro o historial de registro, en castellano) al rastreo secuencial en un archivo o en una base de datos de los eventos de un proceso cualquiera, ya sea una aplicación, actividad, entre otros. De esta forma se conoce el comportamiento del sistema pudiendo solucionar problemas determinados en cada caso. Se puede llamar al proceso de generación de log como acciones más similares para todos, como guardar o registrar. (7)

Cada vez que se utiliza un dispositivo conectado a una red, ya sea ordenador, Tablet o su propio teléfono móvil, se ejecutan numerosos procesos los cuales no son perceptibles por el usuario. Mientras todo funciona perfectamente el usuario está contento y la compañía del soporte y

mantenimiento también, el problema llega cuando se presentan problemas, y se producen errores o quieres saber más información de las acciones que se producen ya sea para aligerar el producto o para hacerlo más sofisticado, en estos casos se debe acceder a los llamados archivos de registro. Estos registros suelen ser gestionados automáticamente por todas las aplicaciones, servidores, bases de datos y sistemas, los cuales nos permiten controlar de forma centralizada todos los procesos relevantes. (7)

En la mayoría de las ocasiones los ficheros registros no se consultan frecuentemente, ya que cumplen con su función, que son visualizados y monitorizados en caso de un funcionamiento inesperado del producto. Este fallo de funcionamiento puede ser causado por errores producidos por el programa o sistema, pero también hay una particularidad muy importante que es el error producido por el comportamiento de los usuarios, esto resulta muy importante para saber y dejar registrado quienes y desde donde se puede acceder a la información de por ejemplo un registro de un servidor web. (7)

Los registros son archivos normales de texto, simplemente como su nombre indica son ficheros que registran todo tipo de acciones que se producen en los procesos que han sido relevantes por el programador para un visión del funcionamiento de su sistema. Un ejemplo muy usual son los archivos registros de una base de datos o de una máquina virtual, donde se registran todos los cambios de las transacciones completadas exitosamente, dando a conocer por ejemplo el tiempo utilizado y transacciones simultaneas. También será útil para cuando llegue dentro de estos datos un error, el cual lo detectará automáticamente y actuará con el cómo el propio informático haya decidido. (7)

1.3.1. Trabajo con Logs

La gran suerte de trabajar con ficheros registros es que son fichero ASCII, con lo cual su gestión es sencilla. Se debe de recordar que un fichero de registros deja constancia de las transacciones del producto, y dadas las características del protocolo HTTP (no existe el concepto de sesión), cada línea del fichero corresponderá a una petición realizada al servidor por parte de un cliente. Como gran problema tenemos que un archivo de registros tiene un gran tamaño, no de peso, si no de extensión, por eso es necesario pre procesarlos para sacar datos de ellos, para ello se suele optimizar el archivo ya que muchas líneas del fichero no son relevantes. (7)

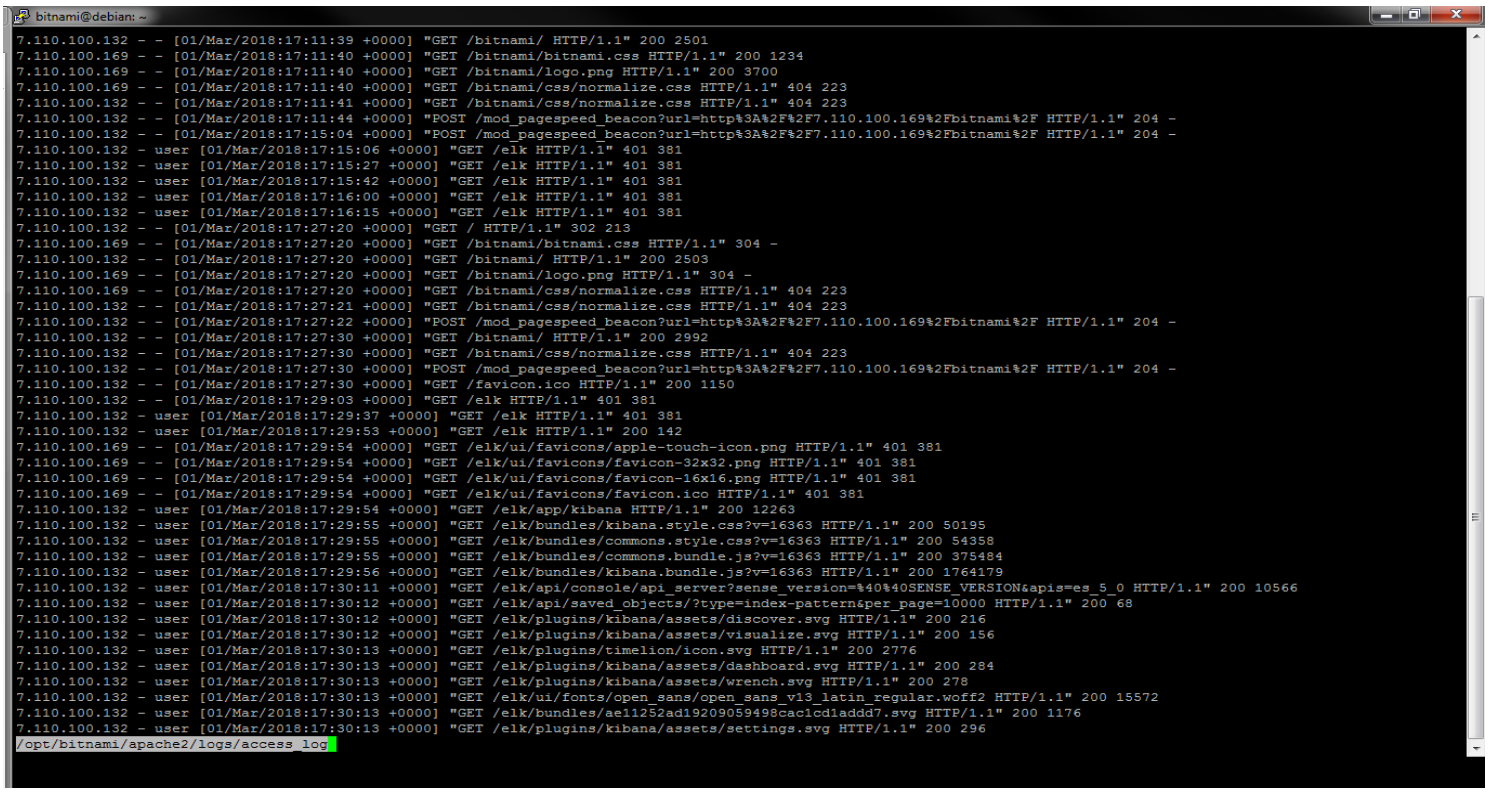

Ilustración 1 Captura de ingesta de logs

Un ejemplo capturado en la instantánea anterior es este:

7.110.100.133 user[01/Mar/2018:17:29:56+0000]"GET/elk/bundles/kibana.bundle.js?v=16363 HTTP/1.1" 2001764179

Este es un registro en una máquina virtual, como se puede observar la información que proporciona es muy útil, mostrando desde la fecha y hora en la cual se ha realizado la llamada a la máquina, la IP del terminal desde donde se ha realizado la llamada, el tipo de acción que se ha realizado en este caso un GET, que se emplea en http para obtener información, trayendo información que se encuentra en el servidor en forma de archivos, bases de datos. Una línea de registro puede obtener mucha más información, como la confirmación de la llamada (con un 200), el estado de la máquina, o simplemente mensajes de tipo INFO o ERROR para identificar de qué tipo de registro es a simple vista. (7)

1.3.2. Tipos de Logs

Existe un gran número de tipos de registros, muchos por su nombre son propios de determinados sistemas operativos, entre los más conocidos están los correspondientes al Sistema, Aplicaciones y Seguridad que son nativos del sistema operativo Windows, donde a los registros de eventos se les conoce como *EventLogs*, mientras que los del *Kernel* corresponden a los sistemas operativos UNIX. En UNIX los eventos se les conocen como Mensajes *Syslog*

debido al protocolo o programa de mismo nombre que traen estos sistemas operativos para la monitorización de los sucesos ocurridos en el sistema. (8) Otros tipos de logs son:

-Del Kernel: los que son del núcleo.

-De Correo: generados por los servidores de correo.

-De Web: son generados por los servidores web a partir de los accesos de los usuarios a sussitios.

-De Squid: los servidores proxy generan logs sobre todo el tráfico a través de ellos.

-Y otros.

En el mundo del almacenamiento de información y revisión de registros, el programador es una pieza muy importante a la hora de la creación de estos, ya que este puede diferenciar el tipo notificación de registro según su función. Esto servirá para que el arquitecto software tenga más limitado el número de registros que dejan las aplicaciones, y se podrá hacer un filtro más rápido y acertado a la hora de consultar. Por ejemplo, con la búsqueda de errores, el arquitecto basta con 'machear' los registros de tipo ERROR. (7)

A continuación se pueden ver algunos tipos de notificacionesmás utilizados en los registros de un proyecto de arquitectura software:

Tipo de notificaciones de registro	Funcionalidad
INFO	Mensaje informativo
ERROR	Hay condiciones de error
WARNING	Hay condiciones de advertencia
NOTICE	Condición normal, pero significativa
CRIT	Las condiciones son críticas
ALERT	Se debe tomar una acción correctora inmediatamente
EMERG	El sistema esta inutilizable
ALERT	Se debe tomar una decisión una acción correctora inmediatamente
DEBUG	Mensaje de depuración
GET	Información proporcionada por el servidor
POST	Datos corporales del servidor

Tabla 1 Tipos de notificaciones de los logs más utilizados por un arquitecto software

1.4. Análisis de otras soluciones existentes

Para la argumentación de este epígrafe se realizó una investigación acerca de los sistemas informáticosexistentes a nivel internacional y nacional que se encargan de la centralización de registro paracualquier aplicación informática. Explicando las características y los objetivos fundamentales que engloba,mediante una valoración de las funcionalidades principales que realizan y de los eventos que contienen,se especifica si pueden ser utilizados dichos sistemas para darle solución a la problemática existente enDigilante. Mostrando cómo se comportan a nivel internacional y a nivel de instituciones los sistemasrelacionados con la gestión de registros y la visualización de los datos contenidos en dichos registros. Se muestra también los diversos productos por la plataforma en que se implementó,buscando describir principalmente las aplicaciones que se han realizados en software libre.

Estos son ejemplos de las soluciones informáticas:

SOLARWINDS

Un sistema informático basado en la colección de registros, análisis y correlación en tiempo real.SolarWinds y Event Manager (LEM) proporciona registro fácil y potente, mediante la automatizacióninteligente y la gestión de eventos desde cualquier lugar donde se generan los datos para las operacionesde seguridad y protección. El producto permite solucionar de manera eficaz los problemas de rendimientoy seguridad donde se aplique, generando un volumen considerable de registros mediante la correlación deeventos múltiples. Sus funcionalidades permiten aplicar dicho producto en entornos con autonomía en suscentros de datos y tenga necesidad de contener un alto nivel de visibilidad, recurriendo a losalmacenamientos de largo plazo y los requisitos de archivo. SolarWinds y Event Manager (LEM) leproporciona la visibilidad y la protección que necesita para la seguridad y el cumplimiento y protección desus datos y de la intimidad de los clientes en cuestión de horas.(10)

WHATSUP LOG MANAGEMENT

Un sistema informático que permite recoger, almacenar, archivar, analizar e informar sobre Syslog,registros de eventos de Windows, o W3C registros generados por servidores de aplicaciones Web,balanceadores de carga, firewalls, servidores proxy o aparatos de seguridad de contenido. Herramientaque permite el almacenamiento de sucesos, filtrar, analizar e informar sobre los datos de registros paraverificar el éxito de las políticas de seguridad interna y demostrar el cumplimiento regulatorio, generandoinformes centrados en el cumplimiento de reglas oficiales de personal, de seguridad y cumplimiento. Losarchivos de registro (Syslog, sucesos Windows, registros W3C) contienen información crítica que podríareducir la exposición de la organización a los intrusos, malware, daños, pérdidas y responsabilidadeslegales.(11)

KD Reports

Un sistema informático que permite generar informes o reportes imprimibles de códigos y descripcionesXML. Los informes pueden contener párrafos de texto, tablas, encabezados, pies de página y mucho más;permite desde la visualización de los contenidos de bases de datos a la creación de facturas odocumentos de identidad. Se trata de una herramienta de desarrollo utilizada en el código fuente, peropermite el uso de plantillas que se crean por el personal de diseño. De esta manera, la actualización deinformes no requiere necesariamente cambios de código. KD Reports es capaz de recuperar los datos delos informes de bases de datos. Además, interactúa con Qt Modelo-Vista tecnología para acceder a losdatos existentes sobre aplicaciones del modelo. Los informes creados se pueden mostrar en un cuadro dediálogo de vista previa que forma parte de los informes de KD. Ellos pueden ser guardados en archivosPDF o ser enviados directamente a una impresora (12), dicha aplicación está implementada en software libre.

TRIPWIRE LOG CENTER

Tripwire log center es un registro inteligente dedicado a la solución de la seguridad y la gestión de eventoscon las capacidades como: la vigilancia de las amenazas en tiempo real y control de configuraciónintegrada del sistema a que se aplica, creando registros y analizándolos según un patrón definido por laaplicación al que se le integra, para así detectar según datos almacenados posibles violaciones en laseguridad y en las configuraciones pertinentes. El producto brinda a los administradores la posibilidad deinvestigar los problemas y supervisar el sistema y disponibilidad de las aplicaciones con capacidades degestión de registros que te ayudarán a descubrir la existencia de algún fallo. Tripwire log center alerta entiempo real cuando el registro de los sistemas críticos se apaga, ya sea accidentalmente o con intenciónmaliciosa. También se puede utilizar para determinar la causa raíz de las investigaciones forenses deseguridad.(13)

Sistema	Gestión de registros	Emisión de alertas	Generar reportes en formato PDF	Monitoreo en tiempo real	Fácil de usar y entender	Acceso centralizado para el análisis
SOLARWINDS	Sí	Sí	Sí	Sí	Sí	No
WHATSUP LOG MANAGEMENT	Sí	Sí	Sí	Sí	Sí	No

KD Reports	Sí	No	Sí	No	Sí	No
TRIPWIRE LOG CENTER	Sí	Sí	Sí	No	Sí	Sí

Ilustración 2: Análisis de otras soluciones existentes

Lossistemasinformáticos mencionados utilizan mayormente sus funcionalidades para los registros de datos con referencia a laseguridad y protección de los sistemas al que se integra. Estos fueron creados con las características propias de cada compañíao institución que los implementó, además de responder a ciertas disposiciones que justifican el propósitode su creación, tales como: el objetivo especificado por un mercado, las herramientas que pueden usar, laestructura definida y en algunos casos el lenguaje en que se programa, que son privativos, además estos están limitados porespecificaciones propias de cada compañía que los creó.Por lo que se puede decir que es poco factibleadquirir una de estas aplicaciones que están desarrolladas en otro esquema que habría que adaptar a lascaracterísticas del sistema Digilante, sistema queposee diversas característicaspropias que lo diferencia de los restantes productos de video vigilancia o de cualquier otro tipo.La estructura definida en la implementación de Digilante exige que el componente generador de registros se implemente de acuerdo a las funcionalidades que englobe y a la necesidad definida por losadministradores de dicho sistema.

1.5. Herramienta para el monitoreo de registros

Existen en el mundo disímiles herramientas para la monitorización de registros, entre las más utilizadas se encuentran: Open Source,EventLogAnalyzer, PandoraFMS, Kibana, entre otras, de las cuales el equipo de desarrollo desea utilizar la herramienta Kibana.

AL escogerse Kibana como herramienta de visualización, es imperativa la utilización del stack ELK en eltrabajo, por lo que además de Kibana se usará Elasticsearch y Logstash. Esto hace necesario un estudiode estas herramientas.

1.5.1. ELK

ELK(*ElasticSearch, Logstash y Kibana*) es un servidor que gestiona sus propios registros, los cuales nos muestran las llamadas que tiene este servidor. ELK ingesta datos con un *ElasticSearch* procedente de archivos de registros, y los muestra y monitoriza en un Kibana. (7)

Principales usos de ELK:

Cada vez son más las grandes compañías que usan estas herramientas para la gestión y servicio de sus aplicaciones, los usos pueden ser muy diversos, aunque se pueden centralizar en estos:

-Almacenamiento de *registros*: Guardan los rastros creados por las aplicaciones vinculadas a la plataforma, con una posible gestión de estos a posteriori, se pueden guardar registros durante meses, incluso durante años, como ocurre en muchas entidades, en ocasiones por normas regidas de política de empresa.

-Trazabilidad de extremo a extremo: Ayuda al movimiento de información desde el inicio hasta la posible gestión de esta. Siempre pudiendo gestionar esta, independientemente del lugar donde se encuentre.

-Custodia y fuscación de registros: Da valor a los simples registros de registros proporcionados por otras fuentes de datos, son asignados y desarrollados por la plataforma, almacenándolos y registrándolos para un uso posterior del cliente.

-ForensticAnalytics: Ayuda a la examinación de los datos estructurados en contra de incidente y delitos financieros. Este objetivo ayuda a descubrir y analizar patrones de actividades fraudulentas. (7)

1.5.1.1. Elasticsearch

Elasticsearch es el gran pilar fundamental de la aplicación ELK, es una herramienta servidor basada en Lucense. Está formada por un potente motor de búsqueda de texto completo, distribuido y con capacidad de realizar varias tareas a la vez. ElasticSearch está compuesta por un interfaz web basado en APIRest y cuyos documentos están en lenguaje JSON. (7)

ElasticSearch es una potente herramienta la cual permite indexar e ingestar un gran volumen de datos, con los cuales posteriormente se harán consultas sobre ellos, estas múltiples búsquedas pueden ser aproximadas, resaltadas y a tiempo real. Uno de los usos más comunes es hacer consultas a texto completo, al estar los datos indexados, los resultados que se buscan se obtendrían de forma muy rápida, también permite hacer una serie de cosas que no nos permite una base de datos convencional, aunque soporta el lenguaje SQL de búsqueda a texto completo, también permite hacer búsquedas con diferentes analizadores según por ejemplo el idioma de la propiedad en que se busque. (7)

Elastic, como se conoce dentro del argot de la arquitectura software, usa sus propios conceptos y aunque no es una base de datos relacional (es una base de datos NoSQL) algunos pueden ser similares a las bases de datos convencionales. Lo que en una base de datos relacional es un esquema en ElasticSearch es un índice, lo que en la primera es una tabla en ElasticSearch es un tipo, una fila es un documento y finalmente una columna es una propiedad respectivamente. (7)

1.5.1.2. Logstash

Logstash es una herramienta para la administración de registros. Esta herramienta se puede utilizar para recolectar, parsear y guardar los logs para futuras búsquedas. Esta aplicación se encuentra basada en iRuby y requiere de Java Virtual Machine para correr.

Esta herramienta está basada en el funcionamiento de la integración de entradas, códec, filtros y salidas. Las entradas son las fuentes de datos que se usarán posteriormente: los códec convierten un formato de entrada en otro que Logstash acepte y este último formato en otro de salida. Los códec se usan normalmente cuando los datos no son de texto plano. Los filtros son acciones que se utilizan para procesar en los eventos y permiten modificarlos o eliminar eventos para luego ser procesados. Finalmente, las salidas son los destinos donde los datos procesados son derivados.

Logstash puede ser configurado para utilizar múltiples servidores, pero solo enviarás los registros a uno de ellos hasta que este falle. Si sucede este error, todos los registros previamente recolectados serán accesibles hasta que el host donde se aloja el Logstash este habilitado. Logstash soporta un servidor corriendo como máster y servidores en espera. (7)

Funciones de Logstash:

- Centralizar el tratamiento de todos los tipos de datos: Se puede decir que Logstash es una tubería de datos que le ayuda a procesar los registros y a otros datos de eventos de una gran variedad de sistemas. Logstash puede estar conectado a una gran variedad de fuentes.

-Normalización de varios esquemas: Logstash permite analizar estos datos y convertirlos en un formato común antes de insertarlo en la base datos.

-Extender a formatos de registro personalizados: La mayoría de los registros escritos de las aplicaciones tienen formatos personalizados. Logstash proporciona una forma rápida y cómoda para analizar estos registros a escala.

-Añadir Plugin de otras fuentes: Logstash proporciona una API para el desarrollo de plugin. Los propios usuarios son los que pueden realizar y desarrollar estos plugin para la mejora de la plataforma. (7)

1.5.1.3. Kibana

Kibana es una herramienta de exploración y visualización de datos de código abierto que se utiliza para el análisis de registro y series de tiempo, monitoreo de aplicaciones y casos de uso de inteligencia operacional. Kibana ofrece una estrecha integración con Elasticsearch, que convierte a Kibana en la opción predeterminada para visualizar los datos almacenados. Kibana también es popular debido a sus características potentes y fáciles de usar, como histogramas, gráficos de líneas, gráficos circulares, mapas de calor y soporte geoespacial incorporado.

Kibana ofrece gráficos intuitivos e informes que puede utilizar para navegar de forma interactiva en grandes cantidades de datos de registro. Puede arrastrar dinámicamente ventanas de tiempo, acercarse a subconjuntos de datos específicos, alejarse para ver una imagen más grande y profundizar en los informes para extraer información procesable de sus datos. (7)

Las principales ventajas de esta herramienta son:

-Ofrece analíticas flexibles y plataforma de visualización de datos.

-Resumen en tiempo real y gráficos de flujo de datos.

-Interfaz intuitiva para todo tipo de usuarios.

-Permite compartir fácilmente la información, las gráficas de datos y cuadros de mando.

Gracias a estas herramientas puedes acceder a los datos importantes para el negocio, independientemente de donde se encuentren y en que formato. Y lo que es más importante, visualizarlos de una forma fácil que te ayudará a explorar los datos más valiosos de la empresa. (7)

1.6. Metodología de desarrollo de software

Las metodologías de desarrollo de software son indispensables para crear o actualizar software de calidad que cumpla con los requisitos de los usuarios; son una parte fundamental de la Ingeniería de software la cual denomina metodología a un conjunto de métodos coherentes y relacionados por unos principios comunes. (9)

En el campo del desarrollo de software, existen dos grupos de metodologías, las denominadas tradicionales (formales) y las ágiles. Las primeras son un tanto rígidas, exigen una documentación exhaustiva y se centran en cumplir con el plan del proyecto definido totalmente en la fase inicial del desarrollo del mismo; mientras que la segunda enfatiza el esfuerzo en la capacidad de respuesta a los cambios, las habilidades del equipo y mantener una buena relación con el usuario. (10) Entre las metodologías ágiles, las más utilizadas en nuestros días son MSF, XP, SCRUMP y SXP. Por otro lado las metodologías tradicionales o pesadas más manipuladas son: OPEN, METRICA 3 y RUP.

Se asume la metodología Prodesoft porque la propuesta de solución es un módulo para una herramienta informática existente y la metodología que utiliza el equipo de desarrollo es precisamente esta.

1.6.1. Metodología Prodesoft

Proceso de Desarrollo y Gestión de Proyectos de Software (PRODESOFT) tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un cliente con una planificación y una estimación de recursos predecibles. Los elementos de un proceso y sus relaciones deben responder Quién debe hacer Qué, Cuándo y Cómo.

Quién: Las personas participantes en el proyecto de desarrollo desempeñando uno o más roles específicos.

Qué: Un artefacto es producido por un rol como resultado del desarrollo de sus actividades. Los artefactos se especifican utilizando notaciones. Las herramientas apoyan la elaboración de artefactos.

Cómo y Cuándo: Las actividades son una serie de pasos que lleva a cabo un rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos artefactos. (11)

Prodesoft concibe el ciclo de vida del proyecto como una composición de 5 fases: inicio, modelación, construcción, explotación experimental y despliegue, donde cada fase terminará en un hito con el objetivo fundamental de evaluar y decidir el paso a la siguiente fase de desarrollo. Su modelo de desarrollo de software describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones con una combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. En las primeras fases las iteraciones son realizadas con mayor énfasis en la determinación del alcance del proyecto, la planificación, identificación y descripción de requisitos y la creación de la línea base de la arquitectura, para luego centrarse en el análisis, diseño, implementación y pruebas de los requisitos funcionales. Las iteraciones sucesivas se construyen sobre los artefactos de desarrollo tal como quedaron al final de la última iteración, lo cual permite reducir los riesgos más significativos para el éxito del proyecto. Su premisa de desarrollo basado en componentes permite alcanzar un mayor nivel de reutilización de software y ejecutar las pruebas de forma individual antes de probar los componentes ya ensamblados. (12)

Ilustración 3: Fases del proceso de desarrollo de la metodología Prodesoft

1.7. Herramientas de ingeniería del software asistida por computadora (CASE)

Las herramientas CASE (*ComputerAided Software Engineering*, Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. A continuación, se enuncian los beneficios de dichas herramientas:

-Permiten la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta consiguen agilizar el trabajo.

-Facilitan la realización de prototipos y el desarrollo conjunto de aplicaciones.

-Simplifican el mantenimiento de los programas.

-Mejoran y estandarizar la documentación.

-Aumentan la portabilidad de las aplicaciones.

-Facilitan la reutilización de componentes software.

-Permiten un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización degráficos. (13)

Existen diversas herramientas CASE que soportan el lenguaje de modelado UML (Lenguaje Unificado de Modelado), entre estas se destacan Rational Rose y Visual Paradigmfor UML. Ambas herramientas son muy potentes y permiten la generación de diversos diagramas como los de clases, de objetos, de casos de uso del negocio y de paquetes, además de generar código a partir de los mismos. Por ser multiplataforma, fácil de usar, proporcionar facilidad de trabajo con modelos UML y guardar todo el modelo en un solo fichero, se selecciona Visual Paradigm para UML en su versión 8.0 para la modelación del proceso de desarrollo de la herramienta propuesta. Además, se tuvo en cuenta que la universidad posee una licencia para su uso y esta es una característica que se debe valorar para su selección.

1.7.1. Visual Paradigmfor UML

Visual Paradigm para el Lenguaje Unificado de Modelado (UML) es una herramienta que soporta casi todo el ciclo de desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite hacer dibujos de todo tipo de diagramas de clases, generar código desde diagramas y genera documentación, se distribuye bajo cuatro tipos de licencias: licencia de un solo asiento (Single seatlicense), licencia flotante (Floatinglicense), licencia de suscripción (Subscriptionlicense) y la licencia académica (Subscriptionlicense).

Las principales características de la herramienta son:

-Soporta aplicaciones web.

-Varios idiomas.

-Generación de código para Java y exportación como HTML (Lenguaje de Marcado de Hipertexto).

-Fácil de instalar y actualizar.

-Compatibilidad entre ediciones (14)

Las ventajas que proporciona Visual Paradigmfor UML son:

-Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.

-Corrección sintáctica. Controla que el modelado con UML sea correcto.

-Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.

-Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.

-Trabajo multiusuario. Permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.

-Reutilización. Facilita la reutilización, ya que es una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.

-Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.

-Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta. (15)

1.8. Lenguaje de programación

El análisis de la bibliografía describe a los lenguajes de programación “como un traductor entre el usuario y el equipo. En lugar de aprender el lenguaje nativo del equipo (conocido como lenguaje máquina), se puede utilizar un lenguaje de programación para dar instrucciones al equipo de un modo que sea más fácil de aprender y entender. (16)

Algunos ejemplos de los lenguajes de programación del lado del servidor que se pueden encontrar son Procesador de Hipertexto o HypertextPre-Processor (PHP), Java Server Pages (JSP), Microsoft Active Server Pages (ASP), entre otros.

Los lenguajes de programación del lado del cliente permiten ejecutar código sin que sea enviado al servidor para ahorrar tiempo, es decir que no necesitan ser tratados directamente por el servidor. Algunos de los ejemplos de lenguajes de programación del lado del cliente son JavaScript, Flash, VBScript, XHTML, HTML, CSS, de los cuales se escogió JavaScript.

1.8.1. PHP

El lenguaje de programación PHP, del inglés HypertextPre-Processor, fue desarrollado puntualmente para diseñar páginas web dinámicas programando scripts del lado del servidor. El lenguaje PHP siempre va incrustado dentro del HTML y generalmente se le relaciona con el uso de servidores Linux. El lenguaje PHP presenta cuatro grandes características:

1. Velocidad: PHP no solo es rápido al ser ejecutado, sino que no genera retrasos en la máquina, por esto no requiere grandes recursos del sistema. PHP se integra muy bien junto a otras aplicaciones, especialmente bajo ambientes Unix.

2. Estabilidad: PHP utiliza su propio sistema de administración de recursos y posee un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

3. Seguridad: PHP maneja distintos niveles de seguridad, estos pueden ser configurados en los archivos de configuración.

4. Simplicidad: Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. Además, PHP dispone de una amplia gama de librerías, y permite la posibilidad de agregarle extensiones. Esto le permite su aplicación en múltiples áreas, tales como encriptado, gráficos, Lenguaje de Marcado Extensible (XML) y otras. (17)

1.8.2. JavaScript

JavaScript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos, posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. (11)

El lenguaje de programación Java Script se utiliza principalmente para hacer mejoras de la interfaz de usuario, para hacer validaciones en los formularios de entrada de datos y generar gráficas para mostrar información, entre otros. Para el uso de Java Script no es necesario compilar los programas para ejecutarlos, los programas que se encuentran escritos en Java Script no necesitan de procesos intermedios para poder probarlos. Este lenguaje es desarrollado y actualizado por SunMicrosystem. Su compatibilidad con gran mayoría de los navegadores modernos lo sitúa como uno de los más utilizados. (18)

1.9. Entorno de desarrollo integrado(IDE)

Los entornos de Desarrollo integrado son programas informáticos que contienen un conjunto de herramientas que son muy útiles por los desarrolladores de software, ya que estos optimizan la tarea de escribir programas, corregirlos y ejecutarlos. Ha sido empaquetado como un programa de aplicación, es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). (19)

Existen varios IDE de múltiples lenguajes tales como Eclipse, Oracle JDeveloper, Codenvy, Microsoft Visual Studio, NetBeans, etc. Para el desarrollo del módulo de recuperación y análisis de registros se define por parte del equipo de desarrollo utilizar Visual Studio Codeversion 1.32.3 como IDE, para ello se tuvieron en cuenta las características que se mencionan en el próximo epígrafe.

1.9.1. Visual Studio Codeversion 1.32.3

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux (Ubuntu, Debian, Fedora, Red Hat) y macOS, siendo por lo tanto multiplataforma, es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario.

Características Principales

Entre sus muchas características incluye soporte para la depuración de código ya sea en PHP como en otros lenguajes de programación; control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Este programa nos sirve para trabajar en infinidad de lenguajes de programación tales como: HTML, CSS, JavaScript, JQuery, PHP, TypeScript, LESS, SCSS, Python, C++, Java. En prácticamente cualquier tipo de lenguaje que se nos ocurra o venga a la mente debido a que posee miles y miles de extensiones, siendo una enorme parte de las mismas gratuitas, más o menos el 90% siendo esta también una de las principales características de este increíble software y es que hay extensiones para todo, hasta para saber la hora, ver las estadísticas de los deportes como fútbol o baloncesto, chatear con un equipo de desarrollo, en fin una infinidad de cosas. (20)

También este programa es personalizable al extremo, por lo que los usuarios pueden cambiar el tema del editor; habiendo temas de todos los gustos y colores con diferentes resaltados de código y sintaxis, permite cambiar los íconos del programa por los que nos dé la gana (todo es dicho anteriormente mediante las extensiones), la fuente del código, el tamaño de la misma, agrandar o reducir la interfaz, cambiar los atajos de teclado y las preferencias. (20)

Visual Studio Code ha pegado fuertemente en la comunidad de programadores y esto se nota en la comunidad que hay detrás desarrollando extensiones, temas, así como por las millones de descargas que posee. Su personalización extrema y las miles y miles de extensiones creadas principalmente por los propios programadores y en algunos casos por Microsoft es lo que le dan

vida a este increíble editor de código, permitiéndonos programar en prácticamente cualquier lenguaje y tener casi cualquier funcionalidad agregada mediante la descarga de extensiones o en caso de que no exista lo que buscamos la podemos crear nosotros mismos si sabemos cómo programarlas. (21)

Visual Studio Code a su versión 1.32.3 tiene una única mejora correspondiente con un problema en la carga de directorios con mucha información. (20)

1.10. Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) o DataBaseManagenentSystem (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible. (22)

Existen en el mundo múltiples gestores de base de datos entre los que se encuentran: Microsoft SQL Server, MySQL, SQLite, PostgreSQL

1.10.1. PostgreSQL

Para el desarrollo de la investigación se utilizará el gestor de base datos PostgreSQL, puesto que es un sistema de gestión de bases de datosrelacional libre orientado a objetos, por lo que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. PostgreSQL es considerado el motor de base de datos más avanzado en la actualidad. (23)

Una característica interesante de PostgreSQL es el control de concurrencias multiversión; o MVCC por sus siglas en inglés. Este método agrega una imagen del estado de la base de datos a cada transacción. Esto nos permite hacer transacciones eventualmente consistentes, ofreciéndonos grandes ventajas en el rendimiento. (23)

En PostgreSQL no se requiere usar bloqueos de lectura al realizar una transacción lo que nos brinda una mayor escalabilidad. También PostgreSQL tiene Hot-Standby. Este permite que los clientes hagan búsquedas (sólo de lectura) en los servidores mientras están en modo de recuperación o espera. Así podemos hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema. (23)

PostgreSQL aporta mucha flexibilidad a nuestros proyectos. Por ejemplo, nos permite definir funciones personalizadas por medio de varios lenguajes. Algunos son:

PL/pgSQL

PL/Tcl

PL/Perl

PL/Python

PL/PHP

PL/Ruby

PL/Java

Otra ventaja de PostgreSQL es que está disponible para muchas plataformas y ofrece el código fuente desde el sitio oficial. Algunos de los builds oficiales son:

Mac OS X

Windows

Solaris

Red Hat

Debian

Ubuntu

pgAdmin es la herramienta oficial para administrar nuestras bases de datos en PostgreSQL. Nos permite desde hacer búsquedas SQL hasta desarrollar toda nuestra base de datos de forma muy fácil e intuitiva; directamente desde la interfaz gráfica. (23)

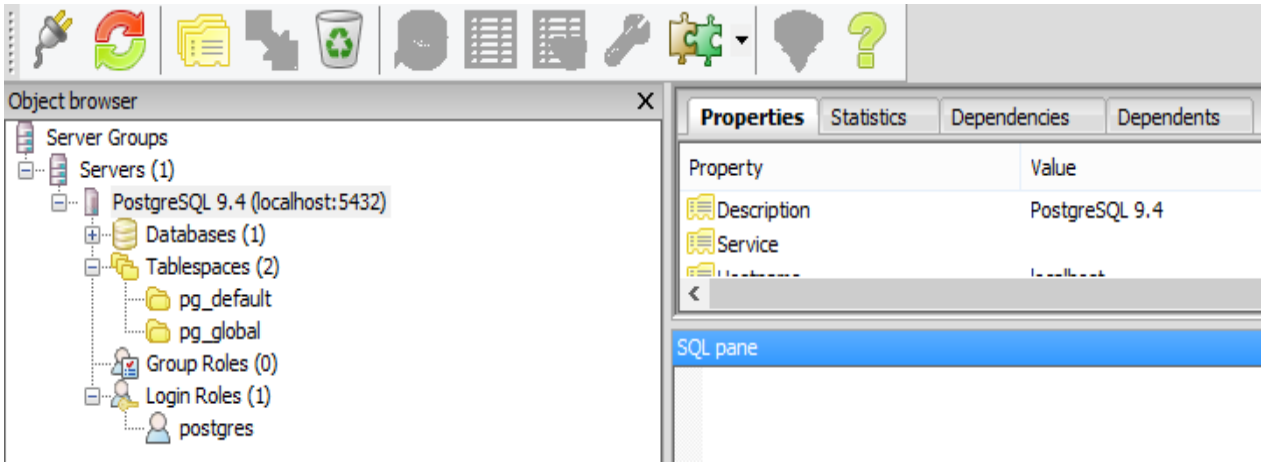

Ilustración 4 Interfaz gráfica del pgAdmin

1.11. Arquitectura de Software

Una arquitectura de software define la forma de trabajar en un sistema, implica definir una solución estructurada que satisfaga todos los requisitos técnicos y operacionales y, a la vez,

optimizar los atributos comunes de calidad como rendimiento, seguridad y capacidad de administración. Además, implica una serie de decisiones basadas en una amplia gama de factores, y cada una de esas decisiones puede tener un considerable impacto sobre la calidad, rendimiento, mantenimiento y éxito general de ese software. (24)

1.11.1. Estilos arquitectónicos

Los Estilos arquitectónicos son un conjunto de reglas de diseño que identifican las clases de componentes y conectores que se pueden utilizar para componer en sistema o subsistema, junto con las restricciones locales o globales de la forma en que la composición se lleva a cabo. (25)

Los estilos se clasifican de la siguiente forma:

Estilo de Flujo de datos

-Tuberías y filtros

Estilos Centrados en datos

-Arquitecturas de pizarra o repositorio

Estilos de Llamada y Retorno

-Modelo-Vista-Controlador (MVC)

-Arquitecturas en capas

-Arquitecturas orientadas a objetos

-Arquitecturas basadas en componentes

Estilos de Código móvil

-Arquitectura de máquinas virtuales

Estilos heterogéneos

-Sistemas de control de procesos

-Arquitecturas basadas en atributos

Estilos Peer-to-Peer

-Arquitecturas basadas en eventos

-Arquitecturas orientadas a servicios

-Arquitecturas basadas en recursos

Para el desarrollo del módulo se asume el estilo arquitectónico Modelo-Vista-Controlador(MVC). Este es un patrón de arquitectura de las aplicaciones software que separa la lógica de negocio

de la interfaz de usuario. Facilita la evolución por separado de ambos aspectos la reutilización y la flexibilidad.

1.11.2. Arquitecturas Modelo-Vista-Controlador

Separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo maneja los datos del sistema y las operaciones asociadas a esos datos. El componente Vista (browser) define y gestiona cómo se presentan los datos al usuario. El componente Controlador (componentes del lado del servidor que manejan los requerimientos de HTTP) dirige la interacción del usuario y pasa estas interacciones a Vista y Modelo. (26)

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. (26)

¿Cuándo se usa?: Se usa cuando existen múltiples formas de ver e interactuar con los datos. También se utiliza al desconocerse los requerimientos futuros para la interacción y presentación.

Ventajas: Permite que los datos cambien de manera independiente de su representación y viceversa. Soporta en diferentes formas la presentación de los mismos datos, y los cambios en una representación se muestran en todos ellos.

Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Desventajas: Puede implicar código adicional y complejidad de código cuando el modelo de datos y las interacciones son simples. (26)

1.12. Marco de trabajo (frameworks)

Un marco de trabajo es una estructura de soporte definida, donde un proyecto de software puede ser organizado y desarrollado. Simplifican el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes, facilitan la programación de aplicaciones, encapsulan operaciones complejas en instrucciones sencillas y proporcionan estructura al código fuente, forzando al desarrollador a crear código legible y más fácil de mantener. Además, permiten facilitar el desarrollo del software, suelen incluir soporte de programas, bibliotecas, y software para desarrollar y unir diferentes componentes de un proyecto de desarrollo de programas. (27)

1.12.1. Zeolides versión 1.1

Zeolides no es más que un conjunto de librerías, herramientas, tecnologías y componentes de software integrados en un marco de trabajo para desarrollar aplicaciones de múltiples propósitos, gran tamaño y grandes volúmenes de datos, que permiten el desarrollo ágil, basado en componentes, centrado en los requerimientos del usuario, las interfaces de usuario y la lógica del negocio de las aplicaciones que con el mismo se desarrollen. Puede utilizarse con múltiples propósitos, como por ejemplo para aplicaciones de tiempo real, su objetivo fundamental es el desarrollo de soluciones web de gestión empresarial. (28)

Características principales que posee:

-Provee una estructura base como plataforma tecnológica para la realización de aplicaciones web.

-Es libre y de código abierto.

-Realizado sobre el funcionamiento del Marco de trabajo Zeolides, sirviendo como interfaz a la utilización del mismo.

-Multiplataforma (Puede ser utilizado sobre Windows, Linux y otros sistemas operativos).

-Permite el desarrollo ágil y basado en componentes, de aplicaciones web empresariales con múltiples propósitos, gran complejidad y grandes volúmenes de datos.

-Centra el desarrollo en el negocio, los requerimientos y las interfaces de usuario.

-Minimiza los tiempos de construcción de los productos de software.

-Maximiza la reutilización del código fuente.

-Estandariza el uso de patrones en las entidades.

-Estandariza las interfaces de usuario en las entidades.

-Disminuye los tiempos de respuesta ante el desconocimiento de la utilización de la plataforma.

-Disminuye los tiempos de respuesta ante situaciones de error en el código fuente.

-Provee a los desarrolladores un entorno de desarrollo seguro.

-Desarrollado en PHP Java Script, con algunos componentes desarrollados en C++, Java y Python.

-Maximiza la ejecución de buenas prácticas de desarrollo de software.

-Fomenta y estandariza el estilo arquitectónico en los proyectos (Utiliza un estilo arquitectónico híbrido resultado de combinar varios estilos y patrones, el Modelo Vista Controlador y el N-Capas).

-Posee una arquitectura basada en componentes y orientada a la nube (web).

-Principales frameworks utilizados: Zeolides, ExtJS y Doctrine.

-Tecnologías utilizadas: Apache, PostgreSQL, RabbitMQ, XMPP, AJAX.

-Utiliza los componentes verticales de Zeolides, Contenedor de inversión de control, Manejador de excepciones y el Tejedor de aspectos arquitectónicos.

-Utiliza el Sistema de Gestión de Seguridad como componente horizontal, para garantizar la seguridad e integridad de datos que guarda.

-Garantiza la integración segura entre componentes y proyectos.

-Garantiza la utilización de tecnologías libres y la integración con las mismas.

-Maximiza la seguridad de los datos y del acceso a los mismos.

-Garantiza la disponibilidad e integridad de los datos.

-Facilita la utilización de información común de forma compartida haciendo su acceso más rápido y fiable.

-Facilita el uso de las estructuras de datos a los desarrolladores.

-Aumenta los niveles de productividad de los proyectos informáticos. (28)

1.13. Servidor Web

La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets y banners). Estos servidores web utilizan el protocolo HTTP (Hypertext Transfer Protocol). Los servidores web están alojados en un ordenador que cuenta con conexión a internet. El servidor

web, se encuentra a la espera de que algún cliente le haga alguna petición, como, por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red. (29)

Algunos ejemplos de servidores web son Nginx, Internet InformationServices (IIS), Cherokee, Tomcat y Apache. En el desarrollo del módulo de recuperación y análisis de registros se decidió utilizar XAMPP.

1.13.1. XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. (30)

Características y Requisitos

XAMPP solamente requiere descargar y ejecutar un archivo .zip, .tar, o .exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas por separado. (30)

1.14 Patrones de diseño

Un patrón de diseño de software se presenta como una solución reutilizable a un problema recurrente en el desarrollo. Su reutilización reduce el esfuerzo, tiempo y costos empleados durante el diseño de software. Los patrones de diseño capturan la estructura y dinámica de una solución que se repite múltiples veces durante el desarrollo de diferentes aplicaciones, generalmente, en un contexto o dominio determinado. (31)

Los patrones de diseño se pueden agrupar en dos grandes grupos; los GRASP (del inglés General ResponsibilityAssignment Software Patterns), que son patrones generales de software para asignación de responsabilidades y los GOF (Gang of Four, en inglés), encargados de la inicialización, agrupación y comunicación de los objetos.

En el capítulo 2 se muestran los patrones de diseño utilizados en la solución informática.

1.15. Conclusiones parciales

El análisis de los sistemas de videovigilancia y de los sistemas de monitoreo de registros existentes en la actualidad, así como características, funcionalidades, herramientas, tecnologías, metodologías, entre otros, constituyen la base teórica para el desarrollo del módulo Monitoreo de Logs del sistema Digilante, permitiendo ampliar los conocimientos para su posterior uso.

Capítulo 2: Análisis, diseño e implementación de la herramienta informática

2.1 Introducción

El presente capítulo expone las principales características del sistema desarrollado, mediante el modelado del negocio y la identificación de los requisitos funcionales y no funcionales. Se describe la arquitectura por la cual se rigió la solución obtenida y los patrones de diseño que fueron utilizados para dar solución al problema que dio origen a la presente investigación.

2.2 Descripción de la solución

El módulo Monitoreo de Logs está diseñado para integrarse al sistema de videovigilanciaDigilante permitiendo llevar a cabo un control de cada uno de los servidores del sistema a través del análisis de los registros generados por estos. De manera general será posible, desde el sistema, tener la supervisión y el control del correcto funcionamiento de los servidores haciendo uso de la plataforma ELK que permite concentrar los registros de múltiples servidores en un único punto, simplificando la labor de gestión del administrador.

Haciendo uso de la API de Kibana se podrá recopilar todos los eventos que ocurren en el sistema Digilante para posteriormente poderlos analizar y visualizar en el módulo Monitoreo de Logs. El proceso de monitorización de los registros se lleva a cabo a través de la herramienta Logstash, la cual recolecta todos los registros para parsearlos y almacenarlos en Elasticsearch, donde además de estar almacenados, también son indexados y enviados a Kibana para finalmente ser leídos y monitorizados.

2.3 Fases de la metodología de desarrollo PRODESOFT

2.3.1 Fase de Inicio

En esta fase se logra una visión preliminar de la problemática a resolver y se definen los recursos relevantes para la ejecución del proyecto. Es decir, se describen los objetivos y el alcance del proyecto, se identifican los involucrados y ejecutores.

Se estima de manera general las actividades a realizar durante todo el ciclo de desarrollo del proyecto, se establece la estrategia a seguir para realizar la modelación del negocio y la captura de requisitos y de ser necesario se estiman los recursos materiales que deberán ser adquiridos.

2.3.2 Fase de modelación

En esta fase se define una visión preliminar de la problemática a resolver por medio de la informatización, los antecedentes de las soluciones informatizadas similares del problema planteado u otras experiencias similares. Se reflejan las generalidades de la propuesta de solución, el tipo de solución a emplear. Además de las principales características técnicas que tendrá la solución definiendo con estos elementos un proyecto técnico. (11)

Se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requerimientos funcionales, obteniéndose la línea base de la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. El hito fundamental de esta fase es la liberación de la arquitectura de sistema, datos y despliegue. (11)

2.3.2.1 Modelo conceptual

El modelo conceptual es una representación visual de los conceptos más significativos en el dominio del problema, identificando los atributos y las asociaciones representándolos en el Lenguaje Unificado de Modelado(UML) como un diagrama de clases. En la fase de inicio se determinó que los procesos del negocio no están claramente definidos, por esta razón se decide representar la situación real del sistema mediante un Modelo de Dominio, facilitando a través de un vocabulario común que ayude a usuarios, clientes, desarrolladores e interesados a comprender el argumento principal del sistema.

2.3.2.1.1 Diagrama del modelo conceptual

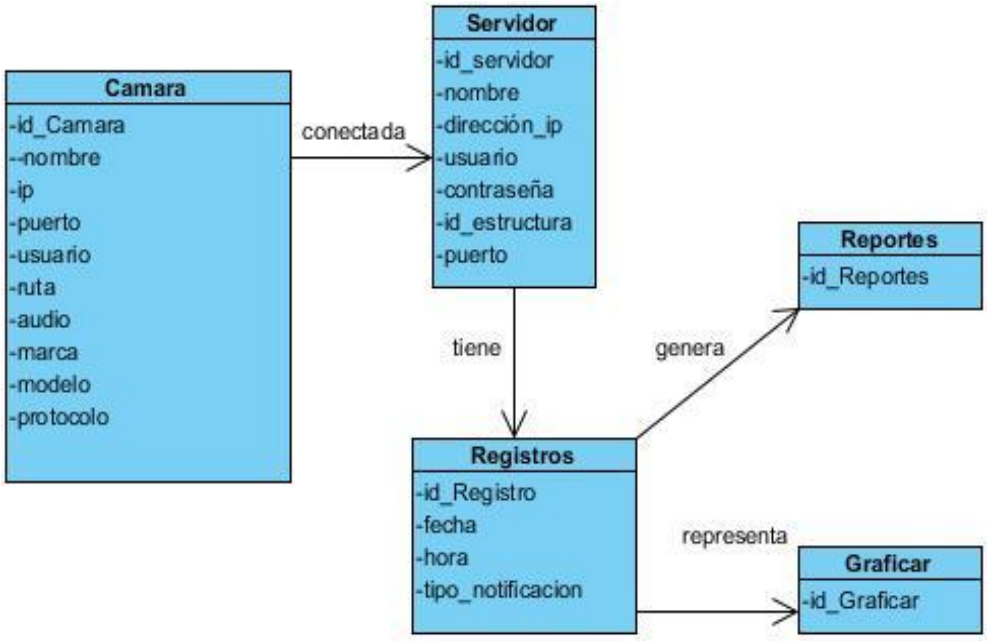

Ilustración 5: Descripción del modelo conceptual

2.3.2.1.2 Descripción de los conceptos relacionados en el modelo conceptual

Concepto	Descripción
Servidor	Ordenador u otro tipo de equipo informático encargado de suministrar información a una serie de clientes, que pueden ser tanto personas como otros dispositivos conectados a él.
Rgistros	Archivos de texto generados por un sistema determinado en respuesta a un evento
Reportes	Eventos que agrupan información de acuerdo con un interés específico, donde se muestran resultados de una búsqueda determinada
Gráfica	Representación a través de figuras o signos
Cámara	Dispositivo para capturar imágenes convirtiéndolas en señales eléctricas, en la mayoría de los casos a señal de vídeo

Tabla 2: Descripción de los conceptos relacionados en el modelo conceptual

2.3.2.2 Definición de requisitos

El propósito de la definición de requisitos es especificar las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento entre el equipo de desarrollo y el especialista funcional, y especificando las necesidades reales de forma que satisfaga sus expectativas. Los requisitos deben ser claros, correctos, inequívocos, específicos, y comprobables. (11)

Los requisitos de software se clasifican en: requisitos funcionales y requisitos no funcionales.

Los **requisitos funcionales** definen las condiciones o capacidades que el sistema será capaz de realizar. Estos describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Estos requisitos, al tiempo que avanza el proyecto de software, se convierten en los algoritmos, la lógica y gran parte del código del sistema. (11)

Los **requisitos no funcionales** son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (11)

2.4.2.1 Requisitos funcionales

No.	Nombre del Requisito	Descripción	Prioridad
RF1	Obtenerdesconexión de cámara del servidor de streaming	Permite llevar un control numérico del funcionamiento del servidor de streaminga través de la cantidad de veces que se ha desconectado la cámara de dicho servidor.	alta
RF2	Obtener desconexión de cámara del servidor de grabaciones	Permite llevar un control numérico del funcionamiento del servidor de grabaciones a través de la cantidad de veces que se ha desconectado la cámara de dicho servidor.	alta
RF3	Obtener desconexión del servidor	Permite llevar un control numérico del funcionamiento de cada uno de los servidores a través de la cantidad de veces que se han desconectado.	alta
RF4	Visualizar datos de registros	Visualiza todos los datos que describen a los sucesos, que se encuentran estructurados dentro de los registros.	media
RF5	Visualizar datos de registros por criterio de búsqueda	Busca en los registros la información de los sucesos y la visualiza por criterios de búsquedas.	media
RF6	Generar reportes	Genera reportes con los datos de los registros, específicamente ficheros de formato pdf.	media
RF7	Graficar desconexión del servidor de streaming	Representa gráficamente el control numérico del funcionamiento del servidor de streaming a través de la cantidad de veces que se ha desconectado la cámara de dicho servidor.	alta
RF8	Graficar desconexión del servidor de grabaciones	Representa gráficamente el control numérico del funcionamiento del servidor de grabaciones a través de la cantidad de veces que se ha desconectado la cámara de dicho servidor.	alta
RF9	Graficar desconexión del servidor	Representa gráficamente el control numérico del funcionamiento de cada uno de los servidores a través de la cantidad de veces que se han desconectado.	alta

Tabla 3 Requisitos funcionales

A continuación, se muestran las especificaciones de un requisito funcional:

Especificación del RF1: Obtener desconexión de cámara del servidor de streaming

Conceptos Tratados	Conceptos	Atributos
	-Servidor de streaming	-No procede
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectada una cámara.	-No procede
Descripción	El usuario debe: 1. Autenticarse en laplataforma Digilante con Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Finalmente seleccionar el servidor de streaming.	
Complejidad	Alta	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador.	
Post-Condiciones	-Se debe mostrar estrictamente los datos del servidor de streaming. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar datos del servidor de streaming.	

Tabla 4 Especificación del RF1: Obtener cantidad de veces que se ha desconectado la cámara del servidor de streaming

2.4.2.2 Requisitos no funcionales

No.	Nombre del requisito no funcional	Atributo de calidad
RNF1	El componente deberá ser usado solamente por los integrantes del proyecto de videovigilanciaDigilante o por los clientes que incluyan el producto a su sistema informático.	Usabilidad
RNF2	La información deberá estar disponible en el período de trabajo definido por los dirigentes del proyecto de videovigilanciaDigilante, limitada solamente por las restricciones de acuerdo a las políticas de seguridad definidas.	
RNF4	Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.	
RNF5	La herramienta para visualizar los logs deberá tener indicadores que permitan conocer al usuario las acciones que debe realizar, por ejemplo botones con íconos sugerentes y alternativa textual.	
RNF6	El sistema se detiene si se ve afectado por ausencia de conexión a través de la red.	Confiabilidad
RNF7	Si el funcionamiento del sistema se ve interrumpido por el fluido eléctrico, una vez que vuelva este, el sistema se reanuda y	

	continúa realizando las operaciones normalmente.	
RNF8	El producto debe ser legible y con colores adecuados, agradables y poco llamativos.	Interfaz
RNF9	El componente deberá estar instalado en una computadora que tenga memoria RAM 512 MB o más, velocidad de procesamiento del microprocesador 1GHz o superior y un disco duro de 80 o más GB de capacidad para poder almacenar el fichero que genera el componente.	Hardware
RNF10	Sistemas operativos: Microsoft Windows 2000/NT o superior, distribución de GNU/Linux.	Software
RNF11	La velocidad de procesamiento de la información será relativamente rápida, no mayor de 20 segundos en cada inserción.	Rendimiento
RNF12	Las restricciones y el tiempo de soporte será establecido por el equipo de desarrollo y los clientes que utilicen el sistema, se recomienda preferentemente 1 día, debido a la importancia que tendrá dicho componente para el sistema Digilante.	Soporte
RNF13	El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesario	
RNF14		Portabilidad
RNF15	Se asignarán los permisos de acceso, escritura, lectura en dependencia del rol que desempeñe cada usuario del sistema.	Seguridad

Tabla 5: Requisitos no funcionales

2.4.3 Diseño de arquitectura de software

Para el desarrollo del Módulo Monitoreo de Logs se define el estilo arquitectónico Modelo-Vista-Controlador (MVC) que es uno de los más utilizados en las aplicaciones web. Este modelo ofrece la ventaja de una Interfaz de Programación de Aplicaciones (API) bien definida; cualquiera que use las API, podrá reemplazar el modelo, la vista o el controlador sin aparenta dificultad; separación clara entre los componentes de un programa; lo cual permite su implementación por separado; conexión dinámica entre el modelo y sus vistas; se produce en tiempo de ejecución, no en tiempo de compilación.

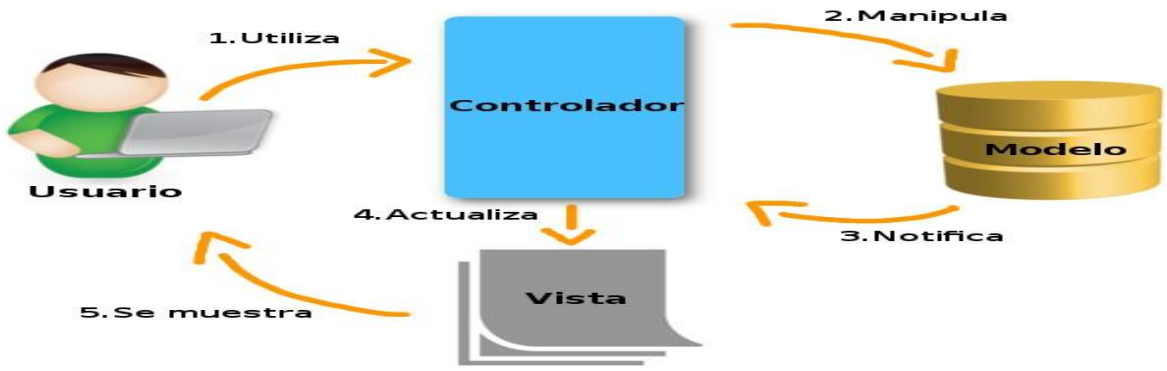

Ilustración 6: Arquitectura MVC

Modelo

La función de esta capa es servir de conector entre el controlador y el gestor de base de datos, es decir es la encarga de manejar los datos asociados al sistema y las operaciones asociadas a esos datos. La idea es manejar las reglas de negocio para acceder la data y cómo se va a manipular. Los modelos se encuentran dentro del paquete “models”.

Vista

Es la capa con la que interactúan directamente los usuarios finales, siendo la encargada de representar los datos del modelo, gestionados por el controlador. En el caso del Módulo Monitoreo de Logs las vistas se encuentran dentro del paquete “views”, las cuales transforman los datos obtenidos del modelo en vistas que son las que permiten al usuario interactuar con la aplicación.

Controlador

Esta capa es la intermediaria entre la vista y el modelo, ya que recibe las solicitudes o peticiones realizadas por el usuario desde la vista, ejecuta las acciones pertinentes y devuelve una respuesta a la interfaz con los resultados de las operaciones realizadas. Para el Módulo Monitoreo de Logs, las clases que se encuentran dentro del paquete “Controller” responden a esta capa, y constituyen la parte de la aplicación que se encarga de realizar una funcionalidad completa y específica.

2.5 Fase de construcción

En la fase de Construcción se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye el producto. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto. (11)

2.5.1 Diseño de interfaz de usuario

Ilustración 7: Interfaz de usuario del sistema

Ilustración 8: Interfaz de usuario del sistema

Ilustración 9: Interfaz de usuario del sistema

Ilustración 10: Interfaz de usuario del sistema

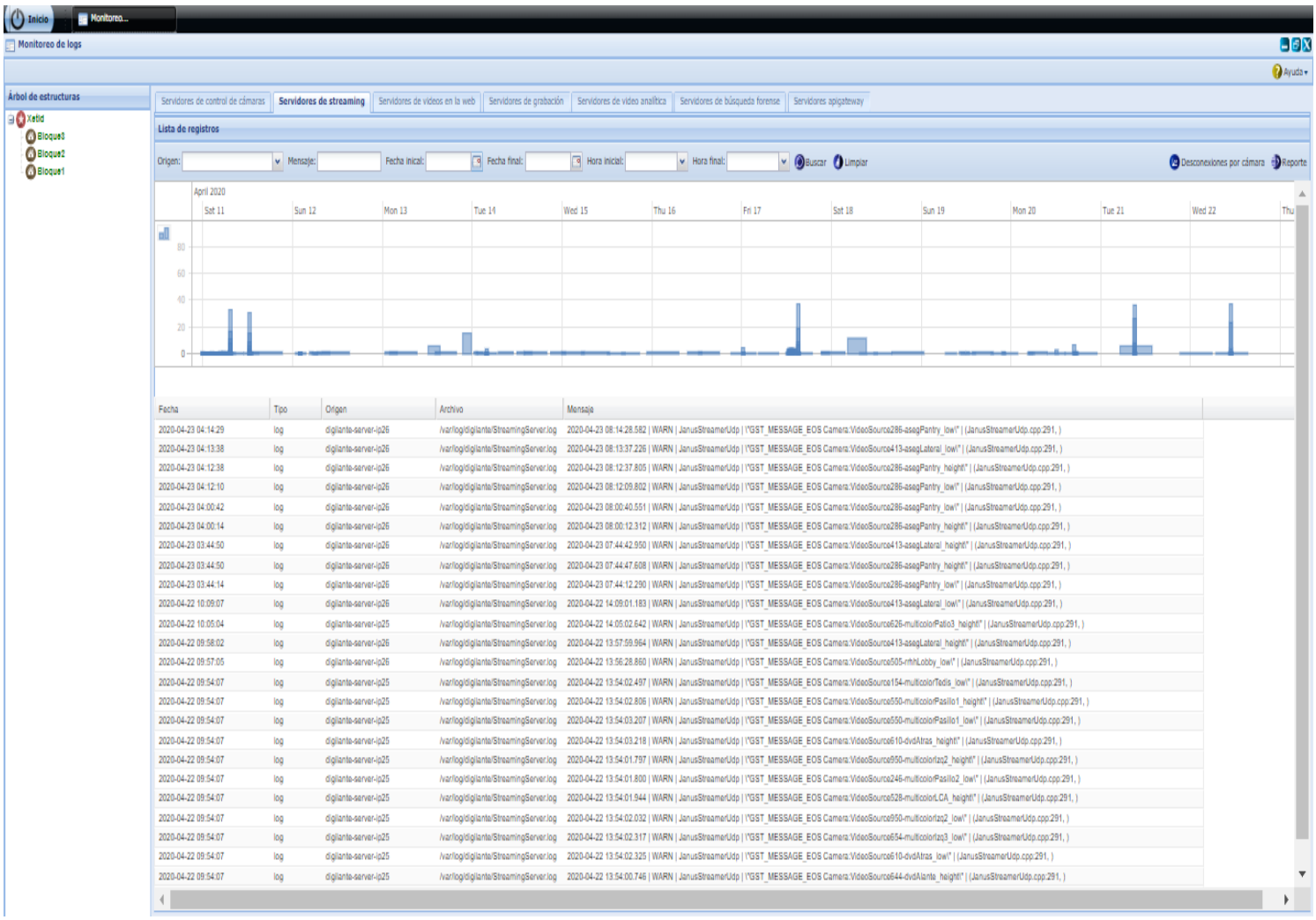

Ilustración 11: Interfaz de usuario del módulo Monitoreo de Logs

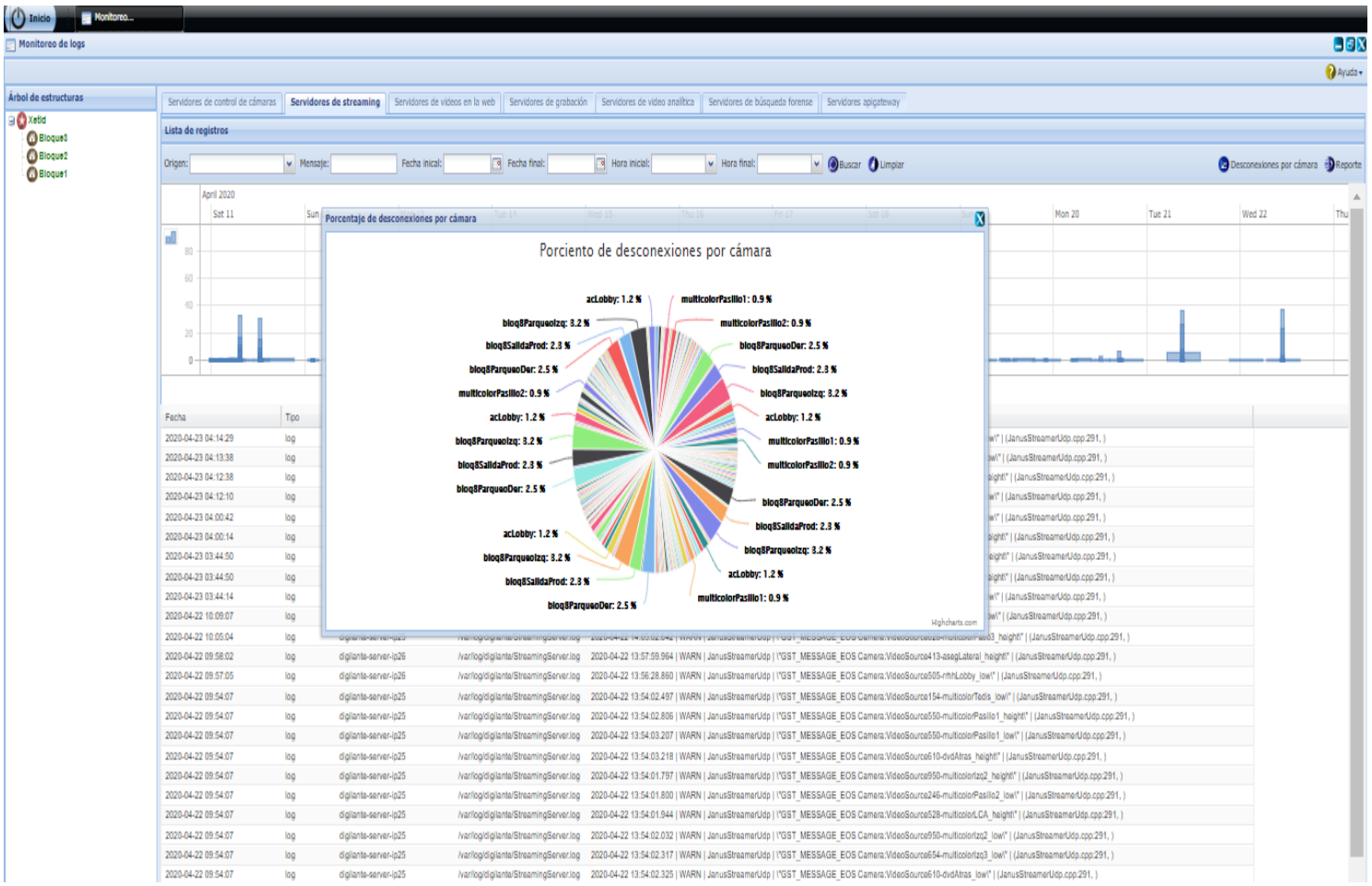

Ilustración 12: Porciento de desconexiones por cámara del servidor streaming

2.5.2.1 Descripción de las interfaces del módulo Monitoreo de Logs

-Cada pestaña se corresponde con un servidor(control de cámaras, streaming, grabación, videos en la web, videoanalítica, búsqueda forense y apigateway).

-Cuando seleccionas una pestaña se muestran los logs correspondientes hasta dos días anterior a la fecha actual.

-Para buscar logs de fechas anteriores debe hacerse mediante los filtros de fecha inicio y fecha fin, así como hora inicio y hora fin.

-Además de los filtros para la fecha y hora, existen otros dos filtros, uno para filtrar el origen de los logs y otro para filtrar en el mensaje del log.

-El botón buscar es para comenzar la búsqueda de logssegún los filtros que se hayan seleccionado.

-El botón limpiar resetea todos los filtros a su estado inicial.

-El botón desconexiones por cámaras abre una ventana con un gráfico de tipo pastel q muestra el porciento de desconexiones por cámaras en una fecha determinada en el servidor seleccionado.

-El botón reporte es para exportar a PDF los logs generados.

-Debajo de los filtros hay una gráfica de barras que muestra la cantidad de logs por fecha

-Debajo de la gráfica de barras esta la tabla de los logs.

2.5.2 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en eldesarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución seaconsiderada un patrón debe poseer ciertas características. Una de ellas es que debe habercomprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es quedebe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño endistintas circunstancias.(34)

Para la propuesta de solución se hizo uso de los Patrones Generales de Software paraasignar Responsabilidad (GRASP) y de los Patrones del grupo de los cuatros: patrones GoF. A continuación, se enumeran los patrones GRASP y GoFutilizados en el módulo Monitoreo de Logs.

2.5.2.1Patrones GRASP

Patrón Alta cohesión: Este patrón tiene como propósito asignar responsabilidades de manera que lacohesión siga siendo alta. En la implementación del Módulo Monitoreo de Logs este patrón sepone de manifiesto en la interrelación que existe entre las clases controladora, las clases gestora ylas funcionalidades del sistema, donde cada una de estas presenta una clase controladora y unagestora, la primera encargada de manejar la lógica de presentación y el flujo de los datosprovenientes de la vista y la segunda encargada de manejar la lógica del negocio de cadafuncionalidad.

Patrón Bajo acoplamiento: Este patrón plantea la baja dependencia que debe existir entre lasclases. En la implementación del módulo Monitoreo de Logs este patrón seutiliza debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización. El patrón se evidencia en cada una de las clases diseñadas para el módulo.

Patrón Controlador: Este patrónpermite asignar la responsabilidad de controlar el flujo deeventos del sistema, a clases específicas, o sea, delega la responsabilidad en otras clases

con lasque mantiene un modelo de alta cohesión, facilitando que se centre todos los métodos en una clase global permitiendo manejar todaslas peticiones con un objeto perteneciente a esta.Este patrón se evidencia en la solución a través de lasclases controladoras que se encuentran en la carpeta Controller.

Patrón Experto: Este patrón consiste en asignar una responsabilidad al experto en información que no es más que la clase que cuenta con la información necesaria para cumplirdicha responsabilidad. Este patrón se pone en prácticaal crear las vistas para mostrárselas alusuario; así como en las clases Controladoras, que son las expertas en información, son lasencargadas de crear los formularios, y a través de los mismos se creará la vista que se le mostrará alusuario.

Patrón Creador: Este patrón se refleja en las clases que tienen la responsabilidad de instanciar objetos de otras clases.

2.5.2.2Patrones GoF

Patrón Singleton (Instancia Única): Este patrón garantiza que una clase solo tengauna instancia, y proporciona un punto de acceso global a ella. Este se pone de manifiesto tantoen las clases controladoras como en las modelo.

2.5.3 Diagrama de clase del diseño

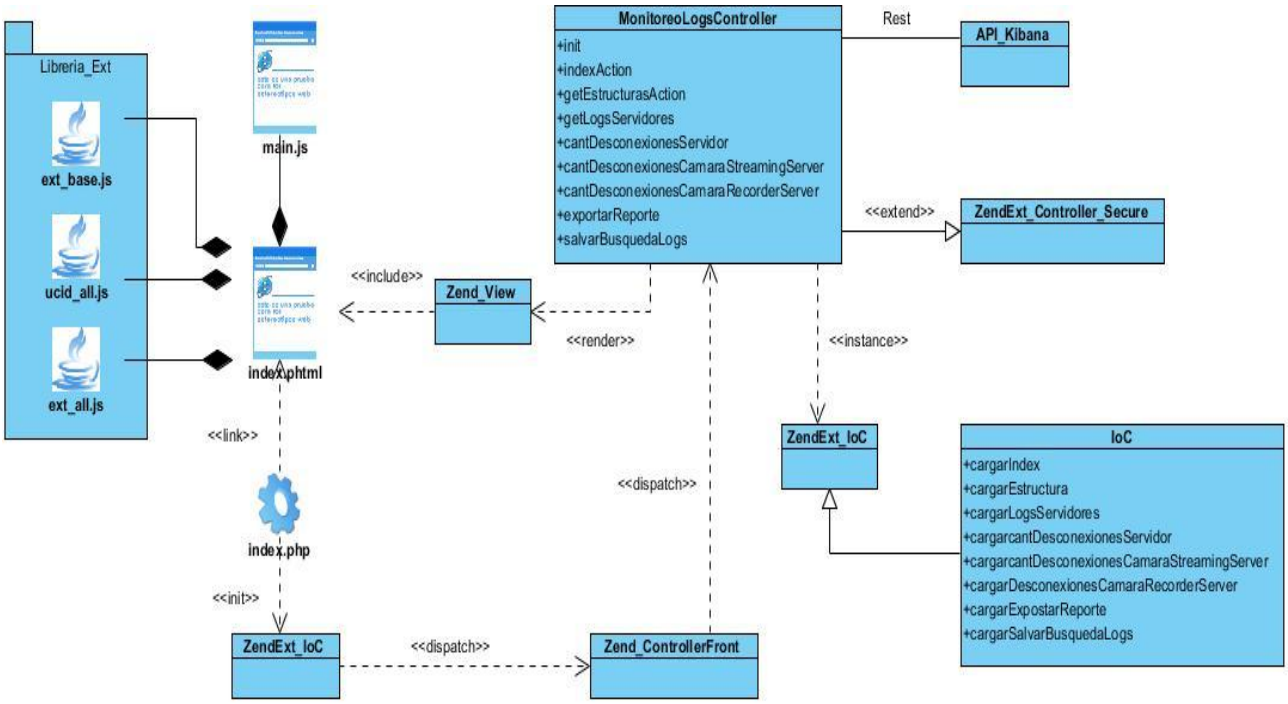

Ilustración 13: Diagrama de clase del diseño.

2.6 Conclusiones parciales

En el presente capítulo se generaron los artefactos de la metodología de desarrollo necesarios para laimplementación de la solución.

Se identificaron nueve requisitos funcionalesy dieciséis requisitos nofuncionales mediante los cuales seestablecen las características de hardware y softwareimprescindibles para el funcionamiento del módulo.

Se utilizó para el desarrollo de la solución el estilo arquitectónico Modelo-Vista-Controlador.El empleo de patrones de diseño GRASP y GoFpara el desarrollo del módulo, permitió que la programación estuviese bien estructurada.Con el uso de la metodología Prodesoft seconstruye un software con un diseño simple pero adecuado.

Capítulo 3: Validación de la propuesta de solución

3.1 Introducción

En el presente capítulo se describe la implementación de la solución mediante el Diagrama deDespliegue.Seproponen las pruebas que verifican el funcionamiento del Módulo Monitoreo de logs del sistema Digilantey se muestranlos resultados obtenidos. De esta manera se comprueba el cumplimiento de los Requisitos Funcionalesdefinidos y el objetivo propuesto para solucionar la problemática.

3.2 Diagrama de despliegue

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que entran en lacomposición de un sistema y el reparto de los programas ejecutables sobre estos nodos.(35)

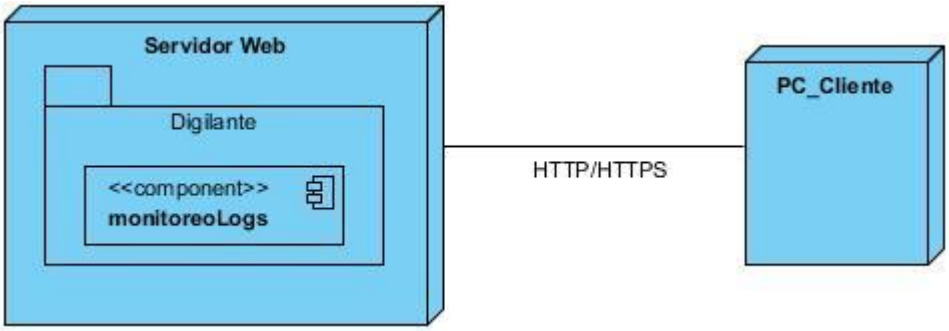

Ilustración 14:Diagrama de Despliegue del Módulo Monitoreo de Logs

En la representación modelada se reflejan dos nodos fundamentales, interconectados mediante el protocoloHTTPo HTTPS. Estos nodos responden a las siguientes descripciones:

-Nodo Servidor Web: Representa el servidor seleccionado para hospedar, ejecutar y mostrar laPlataforma Digilante, con todos los proyectos y plugins que contiene. En el caso particular de lainvestigación se muestra el Componente Monitoreo de Logsdesarrollado.

- Nodo PC Cliente: Representa cada una de las computadoras desde donde los usuarios puedenconectarse al servidor web y acceder a la Plataforma Digilante, paracon los permisosestablecidos,utilizar las funcionalidades del Componente Monitoreo de Logs.

3.3Técnica de validación de los requisitos

Con el objetivo de validar que los requisitos previamente definidos cumplen con las expectativas del cliente, se aplicaron las técnicas de validación de requisitos:

- **Revisiones formales de los requisitos:** se realizaron revisiones formales a cada uno de los requisitos por parte del cliente y del equipo de desarrollo, obteniéndose un total de 4 no conformidades de tipo técnica, de redacción y formato, las que fueron corregidas en tiempo.

- **Construcción de prototipos:** se confeccionaron prototipos no funcionales, dando la posibilidad al cliente de poder comprobar de forma visual cómo quedaría la herramienta, obteniéndose finalmente un prototipo que satisface al cliente.

3.4 Validación del diseño

Un aspecto importante a tener en cuenta en la evaluación del diseño, es la creación de métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objetos. Para validar el diseño realizado se tendrán en cuenta distintos atributos de calidad como: la responsabilidad de las clases, la reutilización, la complejidad de implementación y mantenimiento, el bajo acoplamiento y la cantidad de pruebas unitarias. Las métricas Tamaño Operacional de Clases (TOC) y Relaciones entre Clases (RC) permiten evaluar estos atributos. (36)

3.4.1 Relaciones entre clases

La métrica RC está dada por el número de relaciones de uso de una clase con otra. El primer paso es evaluar un conjunto de atributos de calidad entre los que se encuentran el acoplamiento, complejidad de mantenimiento y reutilización de cada clase. (36)

A continuación, se explican los pasos que se llevaron a cabo para aplicar la métrica:

1. Determinar la cantidad de relaciones de uso (CRU) que poseen las clases a medir.

2. Calcular el promedio de las CRU.

3. Teniendo en cuenta los valores antes obtenidos determinar la incidencia de los atributos de calidad en cada una de las clases, según los criterios expuestos en la tabla siguiente.

Atributos de calidad	Clasificación	Criterio
	Ninguna	CRU = 0

Acoplamiento	Baja	CRU = 1
	Media	CRU = 2
	Alta	CRU > 2
Complejidad de mantenimiento	Baja	CRU <= Promedio
	Media	Promedio< CRU < = 2* promedio
	Alta	CRU > 2* promedio
Reutilización	Baja	CRU > 2* promedio
	Media	Promedio< CRU < = 2* promedio
	Alta	CRU <= Promedio
Cantidad de pruebas	Baja	CRU <= Promedio
	Media	Promedio< CRU < = 2* promedio
	Alta	CRU > 2* promedio

Tabla 6:Criterios de evaluación de la métrica RC

La siguiente tabla muestra un ejemplo del resultado de la aplicación de esta métrica al diseño de laaplicación.

Clase	RC	Acoplamiento	Complejidad de Mantenimiento	Reutilización	Cantidad de pruebas
MonitoreoLogsController	5	Alto	Alta	Baja	Alta
API_Kibana	1	Bajo	Baja	Alta	Baja
ZendExt_Controller_Secure	1	Bajo	Baja	Alta	Baja
IoC	1	Bajo	Baja	Alta	Baja
ZentExt_IoC	2	Medio	Baja	Alta	Baja
Zend_ControllerFront	2	Medio	Baja	Alta	Baja
Zend_View	2	Medio	Baja	Alta	Baja

Tabla 7:Relaciones entre clases

Resultados del instrumento de evaluación de la métrica Relaciones entre Clases

Acoplamiento	Cantidad de Clases	Porciento

Ninguno	0	0
Bajo	3	42.86
Medio	3	42.86
Alto	1	14.29

Tabla 8:Incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

Complejidad de Mantenimiento	Cantidad de Clases	Porciento
Bajo	6	81.71
Medio	0	0
Alto	1	14.29

Tabla 9:Incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento

Reutilización	Cantidad de Clases	Porciento
Bajo	1	14.29
Medio	0	0
Alto	6	81.71

Tabla 10:Incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización

Cantidad de Pruebas	Cantidad de Clases	Porciento
Bajo	6	81.71
Medio	0	0
Alto	1	14.29

Tabla 11:Incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, de manera que los atributos analizados se comportan de la manera siguiente:

Acoplamiento: según los resultados que se muestran, el 42.86% de las clases posee valores bajosde acoplamiento,otro 42.66% son valores medios y el 14.29 es alto, validando una realización correcta del diseño, al no existir un alto valor de dependencia entre clases.

Complejidad de mantenimiento: según los resultados que se muestran en la tabla anterior, el 81.71% de las clases se comportan de forma satisfactoria pues, son de fácil soporte, es decir presentan un una baja complejidad de mantenimiento, el 14.29% restante representa la alta complejidad de mantenimiento.

Reutilización: según los resultados que se mostraron en la figura, el 81.71% de las clases tiene un alto grado de reutilización, el 14.29% representa los valores bajos.

Cantidad de pruebas: luego de aplicar la métrica se obtuvo que el 81.71% de las clases poseen un bajo grado de esfuerzo a la hora de realizar cambios, rectificaciones y pruebas de software, el otro 14.29% restante representa los valores altos.

Según lo analizado anteriormente, los valores de RC se comportan de forma satisfactoria. Estos resultados expresan que las clases del diseño de la herramienta presentan bajo y medio acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas son bajas y en consecuencia el grado de reutilización es alto.

3.5 Pruebas

En todo proceso de desarrollo de software es imprescindible la realización de pruebas para garantizar el buen funcionamiento y la calidad del producto final, así se demuestra el cumplimiento de los objetivos de la investigación.

En la validación de la herramienta se utilizaron los niveles de pruebas de unidad y aceptación, definidos en el Capítulo 1. Para realizar las pruebas de unidad se aplicaron los métodos de caja negra y caja blanca. Las pruebas de aceptación fueron realizadas con el cliente.

3.5.1 Pruebas de unidad

Al desarrollar un nuevo software la primera etapa a considerar es la de las pruebas unitarias, también llamadas pruebas modulares ya que permiten determinar si un módulo del programa está listo y correctamente terminado, las mismas no se deben confundir con las pruebas informales que realiza el programador mientras está desarrollando (37). Como parte de las

pruebas unitarias se aplicaron a la herramienta los métodos de caja blanca y caja negra como se plantea anteriormente.

Pruebas de Caja Blanca:

El método de caja blanca garantiza que se ejerciten por lo menos una vez todos los caminos independientes del código, así como la ejecución de todos los bucles en sus límites operacionales y todas las decisiones lógicas en las vertientes verdaderas y falsas (38).

Pruebas de Caja Negra:

La técnica de diseño de prueba de caja negra es el procedimiento para obtener y/o seleccionar casos de prueba basados en el análisis de la especificación, tanto funcional como no funcional de un componente o sistema sin referencia a su estructura interna (39). A continuación, se presenta el escenario que permite adicionar inversión en mantenimiento.

Descripción	**Variable 1**	**Respuesta del sistema**	**Flujo central**
Permite visualizar los datos de los registros por criterio de búsqueda.	*V (Introducir valores en los campos.)*	*El sistema muestra los registros según los datos insertados.*	*1.Autenticarse en la plataforma Digilante con un usuario valido.* *2. Seleccionar la opción la opción del menú componente Video-vigilancia.* *3. Luego seleccionar el componente Monitoreo y Control.* *4.Posteriormente seleccionar el módulo Monitoreo de Logs.* *5. Seleccionar el servidor deseado.* *6. Introducir valores en los campos* *6. El sistema muestra los registros según los datos insertados.*
	I (introducir un string o carácter extraño en los campos.)	*El sistema no realiza la búsqueda.*	
	I(Dejar campos vacíos)	*El sistema muestra los últimos registros generados por el servidor.*	

Tabla 12: Caso de Prueba del escenario Visualizar datos de registros por criterio de búsqueda

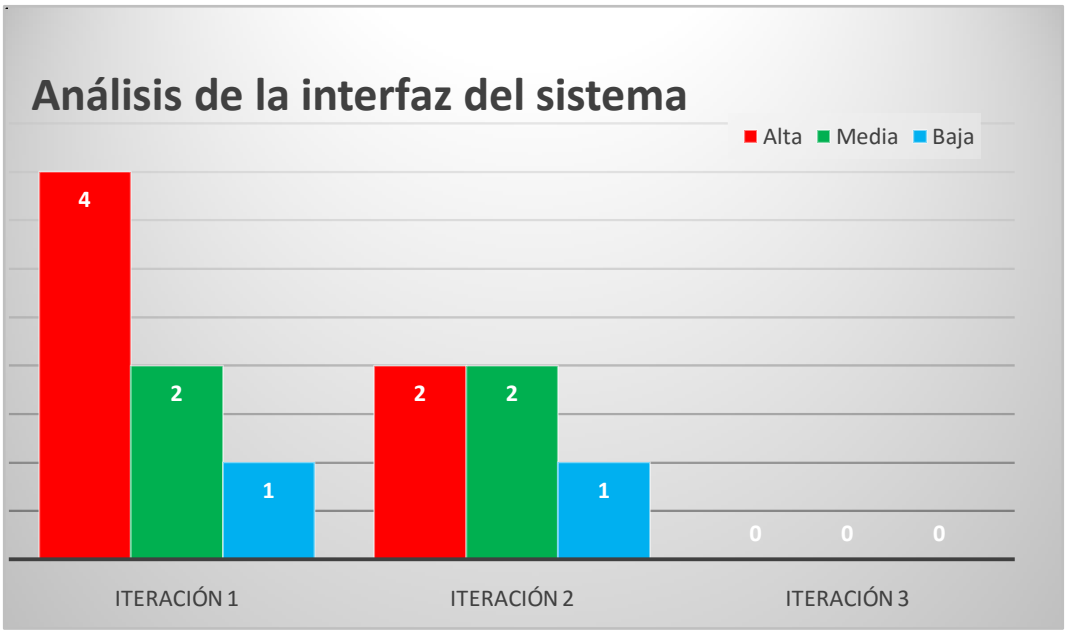

Ilustración 15:Análisis de las pruebas de caja negra

Con el objetivo de comprobar que las funcionalidades de la herramienta se realizaron correctamente y responden a las necesidades del cliente, el método se aplicó en tresiteraciones como se muestra en la **¡Error! No se encuentra el origen de la referencia.**. En la primera se detectaron un total de 7 No Conformidades (NC), predominando entre ellas las de tipo interfaz, al finalizar la iteración todas estas NC quedaron resueltas. En la segunda iteración los resultados mejoraron al disminuir a 2 NC de importancia alta, ya en la tercera iteración se logró resolver las mismas obteniéndose cero NC. La imagen anterior ilustra los resultados de aplicar el método de caja negra, teniendo en cuenta los tipos de NC identificados (interfaz, funcionalidad, redacción).

3.5.2 Pruebas de aceptación

La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar cómo el sistema satisface sus criterios de aceptación, validando los requisitos que han sido levantados para el desarrollo, incluyendo la documentación y procesos de negocio. Está considerada como la fase final del proceso, para crear un producto confiable y apropiado para su uso (40).

Para la aplicación de estas pruebas se confeccionó un caso de prueba de aceptación por cada HU. Seguidamente se muestra el caso de prueba de aceptación de la HU “Visualizar registros por criterio de búsqueda”.

Caso de prueba de aceptación

<table>
<tr><td colspan="2">Código de caso de prueba: 4</td><td>Nombre historia de usuario:Visualizar registros por criterio de búsqueda.</td></tr>
<tr><td colspan="3">Nombre de la persona que realiza el caso de prueba: Ailyn Zulueta González</td></tr>
<tr><td colspan="3">Descripción de la prueba: revisar a través de la herramienta el correcto funcionamiento del RF Visualizar registros por criterio de búsqueda.</td></tr>
<tr><td colspan="3">Condiciones de ejecución: Se deben haber generado registros del servidor.</td></tr>
<tr><td colspan="2">Entrada/Pasos de ejecución</td><td rowspan="2">Resultados esperados:</td></tr>
<tr><td>Acción:</td><td>Entrada</td></tr>
<tr><td>Se introducen datos en los campos. Producción Terminada.</td><td>1. Origen
2. Mensaje
3. Fechainicial
4. Fecha final
5. Hora inicial
6. Hora Final</td><td>El sistema debe mostrar los registros que coinciden con los datos entrados.</td></tr>
<tr><td colspan="3">Evaluación de prueba: Satisfactorio</td></tr>
</table>

Tabla 13:Caso de prueba de aceptación de la HU "Visualizar datos de registros por criterio de búsqueda"

Se realizó un encuentro con el Ing. Rafael Corpas Crehuet, con el objetivo de revisar las funcionalidades de la herramienta, teniendo en cuenta los Casos de Pruebas definidos. Los resultados que se obtuvieron fueron satisfactorios, avalados por el especialista entrevistado, definiendo la aplicación como un importante aporte para el sistema de videovigilanciaDigilante.

3.6 Fase de explotación experimental

Durante la fase de Explotación Experimental se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto. Esta fase solo se ejecuta cuando se tiene como cliente al Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR), por requerimientos que exige este órgano. (11)

3.7 Fase de despliegue

En la fase de Despliegue se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas. (11)

3.8 Conclusiones parciales

La aplicación de técnicas de validación de requisitos, métricas de validación del diseño y pruebas unitarias, certifican la viabilidad del software.

La aplicación de pruebas de aceptación a la solución propuesta, certifican la obtención de un software funcional que responde a los requerimientos del cliente.

Conclusiones

1. El análisis de los sistemas de videovigilancia y de los sistemas de monitoreo de registros existentes en la actualidad, así como características, funcionalidades, herramientas, tecnologías, metodologías, constituyen la base teórica para el desarrollo del módulo Monitoreo de Logs del sistema Digilante, permitiendo ampliar los conocimientos para su posterior uso.

2. El desarrollo delmódulo Monitoreo de logs del sistema Digilantepermite el monitoreo y centralización de los registros de los distintos servidores del sistema.

3. La aplicación de técnicas de validación de requisitos, métricas de validación del diseño, pruebas unitarias y de aceptación a la solución propuesta, certifican la obtención de un software funcional que responde a los requerimientos del cliente.

Recomendaciones

Durante la realización de la presente investigación surgieron ideas que pueden servir como recomendación para el perfeccionamiento del módulo:

- Realizar evaluación de vulnerabilidad por cada servidor del sistema.
- Definir planes de capacidad o cambios arquitectónicos, en caso de ser necesario.

Bibliografía Referenciada

1. *Las TIC, su importancia en al actualidad.* **Licenciada Magíster Tegua de Pepa, Marcela, Doctor García Marino, Carlos y Ingeniero Marianetti, Osvaldo.** 2016.

2. **IMSEL Seuridad, SA.** ***Cámaras de Videovigilancia: elemento clave de los sistemas de videovigilancia-.*** **[En línea]**

3. telemaco. ***La importancia de los logs o registros.*** **[En línea] 24 de 3 de 2019. https://telemaco.es/2019/03/24/la-importancia-de-los-logs-o-registros/.**

4. Martínez Pérez , Raúl y Rodríguez Espanda, Eddy. ***Manual de metodología de la investigación científica.***

5. Bertrán Pompa, Javier Roberto. ***Componente generador de logs y reportes para el sistema de video vigilancia Suria Vision.*** **La Habana : s.n., 2013.**

6. SIPEC WEB. [En línea] 11 de Marzo de 2008. http://sipec.sep.gob.mx/WebHelp/reportes/reporte.htm..

7. S.L., Reviso España Soluciones Cloud. Reviso España Soluciones Cloud S.L. [En línea] 2020. [Citado el:] http://www.reviso.com/Home/General/¿Aún no sabes para qué sirve una API.

8. Valera Torralba, Jesús. ***Diseño e implementación de un sistema de rastreo de registros logs con ELK.*** **Cuenca : s.n., 2018.**

9. López Cardoso, Osmani y García Arévalo, Alexey. ***Propuesta para la recolección centralizada de logs.*** **La Habana : s.n., 2008.**

10. Solarwinds. ***Solarwinds.*** **[En línea] 1 de Diciembre de 2012. http://www.solarwinds.com/.**

11. WhatsUpGold. ***WhatsUp Log Management.*** **[En línea] 28 de Noviembre de 2012. http://www.whatsupgold.com/log-management/tools.aspx.**

12. Head Office, Hagfors, Sweden: Klarälvdalens Datakonsult AB. Kdab. Kdab Reports. [En línea] 2012.

13. Tripwire. *Tripwire.* [En línea] 4 de Enero de 2013. http://www.tripwire.com/it-security-software/log-event-management/log-management/.

14. *Metodologías actuales de desrrollo de software.* Rivas, Carlos Ignacio, y otros. 5980-986, Mexico : Revista Tecnológica de Innovación, 2015, Vol. II.

15. sistemas, División de. *Metodología de desarrollo de software .* Perú : s.n., 2017.

16. XETID. *Proceso de desarrollo de software.* La Habana : s.n.

17. *PRODESOFT-Proceso de Desarrollo y Gestión de Proyectos de.* Softwarew-Defensa, UCID:. La Habana : s.n., 2012, Vol. 15.

18. Valderrama, Johan Sebastián . Herramientas CASE. *MindMeister.* [En línea] 17 de Agosto de 2016. https://www.mindmeister.com/es/742289673/herramientas-case..

19. Visual Paradigm. [En línea] 2018. http://www.visualparadigm.com/support/documents/vpumluserguide/12/13/7572_licensing.html.

20. Visual Paradigm. [En línea] 2014. http://www.ie.inf.uc3m.es/grupo/docencia/reglada/Is1y2/PracticaVP.pdf..

21. MSDN, Microsoft Developer Network. Conceptos básicos: funcionamiento de la programación. [En línea] http://msdn.microsoft.com/eses/library/ms172579%28v=vs.80%29.aspx.

22. Latinoamericana, Red Gráfica. El lenguaje de programación PHP. [En línea] 5 de febrero de 2018. http://redgrafica.com/El-lenguaje-de-programacion-PHP..

23. Eguíluz, Pérez. Introducción a JavaScript. [En línea] http://www.librosweb.es/javascript/..

24. AprendiendoArduino. Aprendiendo Arduino. [En línea] diciembre de 2014. https://aprendiendoarduino.wordpress.com/2016/03/29/entorno-de-.

25. geeks. microsoft anuncia visual studio code v1-32-3. [En línea] 28 de 03 de 2019. https://geeks.ms/jorge/2019/03/28/microsoft-anuncia-visual-studio-code-v1-32-3/.

26. matr1x.cubava. visual-studio-code-el-nuevo-rey-de-los-editores-de-codigo. [En línea] https://matr1x.cubava.cu/visual-studio-code-el-nuevo-rey-de-los-editores-de-codigo.

27. Empresariales, Instituto Europeo de Estudios. revistadigital.inesem. *Informática y Tics.* [En línea] 2016. https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/..

28. platzi. [En línea] 2015. [Citado el: 5 de Diciembre de 2019.] https://platzi.com/blog/que-es-postgresql/.

29. Network., Microsoft Developer. Microsoft Developer Network. *¿Qué es la arquitectura de software? .* [En línea] 2017. https://msdn.microsoft.com/es-es/hh144976.aspx..

30. POSSO, ALCIDES NEPTALÍ RIVERA. repositorio.utn.edu. [En línea] 2010. http://repositorio.utn.edu.ec/bitstream/123456789/988/2/04%20ISC%20164%20Informe%20Te.

31. Sommerville. *Sommerville.* 7ma edición/Cap-6.3.

32. *Marco de trabajo para aplicaciones web de código abiertoen instituciones universitarias.* Recaman Chaux, Hernando y Guerrero Alarcón, Carlos Andrés . Bogotá, D.C. – Colombia : Universidad Cooperativa de Colombia, 2012., 2012, Vols. 10. 2382-4239.

33. XETID. ***Manual de usuario Marco de trabajo para el desarrollo de aplicaciones web. Versión 2.0 Plataforma para el desarrollo Zeolides.*** **La Habana : s.n.**

34. Ibrugor. ***ibrugor.*** **[En línea] 11 de junio de 2014. http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/..**

35. UNID Fondo con valores. ¿ Qué es XAMPP ? ***Plataforma Educativa.*** **[En línea]**

36. Montilva, Jónas A y Ramos, Yajaira. ***Patrones de Diseño para el Modelado de Redes en Sistemas de Información Geográfica.***

37. Hall. ***UML y Patrones.***

38. ***Fundamentos de Ingeniería de Software.*** **Visconti, Marcello y Astudillo, Hérnan. Univercidad Técnica Federico Santa María : s.n., 2004.**

39. ***Bases metodológicas de la investigación.*** **Latorre, A y Del Rincón, D. Y Arnal, J. 2003.**

40. Oré, A. ***UNIT TESTING - PRUEBAS UNITARIAS - CAP 1.*** **2009.**

41. Pressman, R. S. ***Ingeniería del Software. Un enfoque práctico. Sexta Edición.*** **España : Mc Graw Hill, 2003.**

42. Zapata, J. ***Niveles de prueba del software.*** **2013.**

43. Pressman, R. S. ***Ingeniería del Software.*** **España : s.n., 2010.**

44. 10 Sistemas gestores de bases de datos más utilizados en Informática. [En línea] google-10 Sistemas gestores de bases de datos más utilizados en Informática.

45. Zapata, J. ***Niveles de prueba del software.*** **2013.**

46. Pressman, Roger S. ***Ingeniería del Software. Un enfoque práctico. Sexta Edición.*** **España : Mc Graw Hill, 2003.**

47. Arquitectura de Capas. ***migabeta.wordpress.*** **[En línea] 12 de 01 de 2017. https://migabeta.wordpress.com/2017/01/12/arquitectura-de-capas.**

48. introduccion-al-patron-de-arquitectura-por-capas. ***arevalomaria.wordpres.*** **[En línea] 02 de 12 de 2010. https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas.**

49. Fowler, Martin. martinfowler. [En línea] (https://martinfowler.com/articles/microservices.html Artículo de Martin Fowler).

50. redhat. [En línea] [Citado el: 26 de Febrero de 2020.] https://www.redhat.com/es/topics/microservices/what-are-microservices.

51. Vargas, Connie. trycore. [En línea] [Citado el: 26 de Febrero de 2020.] https://trycore.co/buenas-practicas-ti/diferencia-arquitectura-orientada-servicios-microservicios/.

52. powerdata. [En línea] 5 de Agosto de 2017. [Citado el: 26 de Febrero de 2020.] https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-soa-y-cual-es-su-diferencia-con-los-microservicios.

Glosario de términos

1. **CASE**: (ComputerAided Software Engineering), Ingeniería de Software Asistida por Ordenador.
2. **Framework**: Denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes,herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.
3. **GRASP** (del inglés General ResponsibilityAssignment Software Patterns), que son patrones generales de software para asignación de responsabilidades.
4. **GOF** (Gang of Four, en inglés), encargados de la inicialización, agrupación y comunicación de los objetos.
5. **HTML**: (Hyper Text MarkupLanguage) Lenguaje de Marcas de Hipertexto.
6. **HTTP**: (Hypertext Transfer Protocol).
7. **MVC** (en inglés Model – View-Controller): es un patrón de arquitectura de software que separa los datosde una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.
8. **PostgreSQL**: es un sistema de gestión de base de datos relacional orientada a objetos y libre.
9. **SGBD**: (en inglés DataBase Management System, abreviado DBMS). Sistema Gestor de Basesde datos.
10. **Software**: Es un programa o aplicación de que permite a los usuarios el control o realización devarias tareas y que hacen que el trabajo sea más cómodo, rápido y eficiente.
11. **Servidor**: En informática, un servidor es una computadora que, formando parte de una red, proveeservicios a otras computadoras denominadas clientes.

12. **UML**: (UnifiedModelingLanguage) Lenguaje Unificado de Modelado.

Anexos

Especificación del RF1: Obtener cantidad de veces que se ha desconectado la cámara del servidor de streaming

Conceptos Tratados	Conceptos	Atributos
	-Servidor de streaming	-No procede
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectada una cámara.	-No procede
Descripción	El usuario debe: aforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Finalmente seleccionar el servidor de streaming	
Complejidad	Alta	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador.	
Post-Condiciones	-Se debe mostrar estrictamente los datos del servidor de streaming. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar datos del servidor de streaming	

Tabla 14 Especificación del RF1: Obtener cantidad de veces que se ha desconectado la cámara del servidor de streaming

Especificación del RF2: Obtener cantidad de veces que se ha desconectado la cámara del servidor de grabaciones

Conceptos Tratados	Conceptos	Atributos
	-Servidor de grabaciones	-No procede
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectada una cámara.	-No procede
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Finalmente seleccionar el servidor de grabaciones	
Complejidad	Alta	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador.	
Post-Condiciones	-Se debe mostrar estrictamente los datos del servidor de grabaciones. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar datos del servidor de grabaciones	

Tabla 15 Especificación del RF2: Obtener cantidad de veces que se ha desconectado la cámara del servidor de grabaciones

Especificación del RF3: Obtener cantidad de veces que se ha desconectado un servidor

Conceptos Tratados	Conceptos	Atributos
	-Servidor	-No procede
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor.	-No procede
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Finalmente seleccionar el servidor deseado	
Complejidad	Alta	

Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador.
Post-Condiciones	-Se debe mostrar estrictamente los datos del servidor seleccionado. El resultado no debe ser nulo.
Post-Requisitos	Mostrar datos del servidor seleccionado.

Tabla 16 Especificación del RF3: Obtener cantidad de veces que se ha desconectado un servidor

Especificación del RF4: Visualizar los datos que contienen los registros

Conceptos Tratados	Conceptos	Atributos
	-registros	-fecha, hora, tipo de notificación
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor.	-Mostrar atributos de los registros
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Finalmente seleccionar el servidor deseado	
Complejidad	Media	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs en el servidor seleccionado.	
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar registros del servidor seleccionado.	

Tabla 17 Especificación del RF4: Visualizar los datos que contienen los registros

Especificación del RF5: Visualizar los datos de los registros por criterio de búsqueda

Conceptos Tratados	Conceptos	Atributos
	-registros	-fecha, hora, tipo de notificación
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema.	-Mostrar atributos de los registros por

	-Tiene que haber conectado al menos un servidor. -Tiene que haber generados registros	criterio de búsqueda
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Seleccionar el servidor deseado. 6. Finalmente filtrar con los daros deseados.	
Complejidad	Media	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs en el servidor seleccionado.	
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar registros del servidor seleccionado según criterio de búsqueda.	

Tabla 18 Especificación del RF5: Visualizar los datos de los registros por criterio de búsqueda

Especificación del RF6: Generar reportes

Conceptos Tratados	Conceptos	Atributos
	-reportes	
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor. -Estar disponible la librería para JavaScript jsPDF - El sistema debe ser capaz de permitir al usuario seleccionar un registro y generar el reporte en formato pdf	-Mostrar atributos de los registros
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs.	

	5. Seleccionar el servidor deseado. 6.Seleccionar la opción de generar reporte
Complejidad	Media
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs en el servidor seleccionado.
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.
Post-Requisitos	Generar reportes del servidor seleccionado en formato de PDF.

Tabla 19 Especificación del RF6: Generar reportes

Especificación del RF7:Graficar cantidad de veces que se ha desconectado la cámara del servidor de streaming

Conceptos Tratados	Conceptos	Atributos
	-reportes	
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor.	-Mostrar atributos de los registros
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Seleccionar el servidor de streaming. 6. Seleccionar la opción Grafico	
Complejidad	Alta	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs en el servidor streaming.	
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar gráficamente la cantidad de veces que se ha desconectado la cámara del servidor de streaming	

Tabla 20 Especificación del RF7: Graficar cantidad de veces que se ha desconectado la cámara del servidor de streaming

Especificación del RF8: Graficar cantidad de veces que se ha desconectado la cámara del servidor de grabaciones

Conceptos Tratados	Conceptos	Atributos
	-reportes	
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor.	-Mostrar atributos de los registros
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs. 5. Seleccionar el servidor de grabaciones. 6. Seleccionar la opción Grafico	
Complejidad	Alta	
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs en el servidor de grabaciones.	
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.	
Post-Requisitos	Mostrar gráficamente la cantidad de veces que se ha desconectado la cámara del servidor de grabaciones	

Tabla 21 Especificación del RF8: Graficar cantidad de veces que se ha desconectado la cámara del servidor de grabaciones

Especificación del RF9: Graficar cantidad de veces que se ha desconectado un servidor

Conceptos Tratados	Conceptos	Atributos
	-reportes	
Precondiciones	Precondiciones	Pre-requisitos
	-Ser usuario del sistema. -Tiene que haber conectado al menos un servidor.	-Mostrar atributos de los registros
Descripción	El usuario debe: 1. Autenticarse en la plataforma Digilante con un usuario valido. 2. Seleccionar la opción la opción del menú componente Video-vigilancia. 3. Luego seleccionar el componente Monitoreo y Control. 4. Posteriormente seleccionar el módulo Monitoreo de Logs.	

	5. Seleccionar el servidor deseado. 6. Seleccionar la opción Grafico
Complejidad	Alta
Validaciones	-El usuario y la contraseña deben coincidir con algún usuario del sistema Digilante o tener el rol de administrador. -Debe haber generado logs del servidor seleccionado
Post-Condiciones	-Se debe mostrar estrictamente los registros del servidor seleccionado. El resultado no debe ser nulo.
Post-Requisitos	Mostrar gráficamente la cantidad de veces que se ha desconectado la cámara del servidor seleccionado

Tabla 22 Especificación del RF9: Graficar cantidad de veces que se ha desconectado un servidor