UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA. INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de.

Ingeniero de Sistemas

TEMA:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA CENTRALIZADO DE MONITOREO, SUPERVISIÓN Y CONTROL AUTOMÁTICO DE SERVIDORES Y SERVICIOS EN ENTORNOS VIRTUALES DE LA EMPRESA MESSAGE PLUS BASADO EN HERRAMIENTAS DE CÓDIGO ABIERTO.

AUTOR.

LUIS FERNANDO VALLEJO LLUMIQUINGA

TUTOR.

JOSÉ LUIS AGUAYO MORALES

Quito, agosto del 2020

Cesión de derechos de autor

Yo Luis Fernando Vallejo Llumiquinga con documento de identificación N° 1720897915, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA CENTRALIZADO DE MONITOREO, SUPERVISIÓN Y CONTROL AUTOMÁTICO DE SERVIDORES Y SERVICIOS EN ENTORNOS VIRTUALES DE LA EMPRESA MESSAGE PLUS BASADO EN HERRAMIENTAS DE CÓDIGO ABIERTO.", mismo que ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, que dando la Universidad faculta da para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.

Luis Fernando Vallejo Llumiquinga

CI: 1720897915

Quito, agosto de 2020

Declaratoria de coautoría del docente tutor

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA CENTRALIZADO DE MONITOREO, SUPERVISIÓN Y CONTROL AUTOMÁTICO DE SERVIDORES Y SERVICIOS EN ENTORNOS VIRTUALES DE LA EMPRESA MESSAGE PLUS BASADO EN HERRAMIENTAS DE CÓDIGO ABIERTO. Realizado por Luis Fernando Vallejo Llumiquinga, obteniendo un resultado que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto de 2020

…….

Msc. José Luis Aguayo Morales

CI: 1709562597

Dedicatoria

El presente trabajo está dedicado a mis padres, quienes, con su apoyo y constancia en cada aspecto de mi vida, han podido guiarme y darme cada una de las oportunidades que he tenido. Con la finalización de esta etapa de mi vida, espero llenar sus expectativas, y demostrarles como todo su esfuerzo ha servido para hacer a un profesional competente y un ser humano de bien, que cumple con los valores que le inculcaron en su crianza.

Luis Fernando Vallejo

Agradecimientos

A mis padres que han solventado cada una de mis necesidades en todos los momentos de mi educación y de mi vida, les agradezco su esfuerzo, su paciencia y las ganas de verme siendo un profesional y un hombre de valores correctos. A mis abuelos que con su cariño supieron convertirme en un estudiante decidido, a pesar de que muchos ya no se encuentran a mi lado, espero vean que todo ha valido la pena, y que mis pensamientos siempre los van a tener en cuenta, en donde sea que ahora me cuiden.

A la mujer que estuvo a mi lado en grandes momentos de mi carrera, que supo darme una razón más para esforzarme y lograr nuevos objetivos, además de darme consejo y alivio en situaciones que escapaban de mi control, le quiero agradecer por su esfuerzo.

A muchos de los profesores que marcaron mi carrera y me enseñaron como el conocimiento que se adquiere en un aula, se demuestra con el esfuerzo del estudiante en un trabajo, les quiero agradecer y recordar que la educación es el pilar de una sociedad en camino a un futuro mejor.

Finalmente, expreso mi agradecimiento a la empresa MessagePlus, quienes me dieron la oportunidad de experimentar grandes responsabilidades que han enriquecido mi formación como ingeniero de sistemas, y ahora me han encargado el presente proyecto para culminar esta fase en mi carrera universitaria.

Luis Fernando Vallejo

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Resumen

La empresa MessagePlus requiere una plataforma de manejo, control y notificación de incidentes en las terminales que usa para la gestión de sus servicios, estás se alojan en servidores propios bajo el sistema de hosting. La plataforma debe verificar el estado de los servicios que maneja la empresa, mediante el monitoreo de los procesos que los ejecutan, y automatizando algunas de las acciones recurrentes que no necesiten la verificación del personal de soporte.

El uso de Zabbix y Monit, herramientas de código abierto, facilitan el control y monitores de terminales virtuales y estado de procesos respectivamente. Con la implementación de estas herramientas se cubre las necesidades de la empresa desde la gestión de los entornos virtuales, hasta el monitoreo de los servicios principales, junto con el manejo eficiente de notificaciones. Esta plataforma se levanta en la infraestructura propia de la empresa, en una terminal en funcionamiento ininterrumpido. Cuyas métricas se establecen en conjunto con el personal de TI de la organización, quienes tendrán el control total de la plataforma el finalizar el presente proyecto. También se resuelve una solución de monitoreo móvil eficiente y sencilla para el uso del personal.

Ya finalizada la implementación de la plataforma, se tiene acceso completo a la web que engloba Zabbix y Monit, accesible desde cualquier lugar gracias al uso de una IP pública que es proporcionada por el hosting. La web incluye la visualización de paneles detallados del estado, incidentes, y estadísticas de las terminales, así como estado de los procesos monitoreados.

Abstract

The MessagePlus company requires an incident management, control and notification platform in the terminals that it uses to manage its services. They are hosted on their own servers under the hosting system. The platform must verify the status of the services that the company manages, by monitoring the processes that execute them, and automating some of the recurring actions that do not require verification by support personnel.

The use of Zabbix and Monit, open source tools, facilitate the control and monitors of virtual terminals and process status respectively. With the implementation of these tools, the needs of the company are covered from the management of virtual environments to the monitoring of the main services, along with the efficient handling of notifications. This platform is built on the company's own infrastructure, in a terminal in uninterrupted operation. Whose metrics are established in conjunction with the organization's IT staff, who will have full control of the platform at the end of this project. An efficient and simple mobile monitoring solution for staff use is also resolved.

Once the implementation of the platform is finished, you have full access to the web that includes Zabbix and Monit, accessible from anywhere thanks to the use of a public IP that is provided by the hosting. The website includes the display of detailed panels of the status, incidents, and statistics of the terminals, as well as the status of the monitored processes.

Capítulo I

Introducción

La empresa Message Plus con más de 13 años de experiencia, brinda servicios de mensajería masiva, mailing, recargas, servicios IVR y analítica de big data. Con un rack de servidores que brindan estos servicios, en un sistema de disponibilidad 24/7, y ahora desea implementar una plataforma de monitoreo y control para la revisión autónoma de su infraestructura y procesos principales, mientras sea factible. Con una interfaz simple que controle, maneje y solvente problemas que se puedan automatizar.

Las herramientas de código abierto, son lo suficientemente robustas para hacer un sistema de monitoreo y control estable que cumpla con lo requerido en Message Plus resumido en el párrafo anterior. Estas herramientas con capaces de un monitoreo tanto de hardware como de software, todo en una interfaz sencilla para el uso del personal de soporte e involucrados en TI de la organización.

Antecedentes

La gestión de procesos en una empresa es la base del desarrollo de sus aplicaciones, el control y manejo de estos debe ser un eje fundamental para ofrecer un buen servicio, pero a veces la presión y el entorno de desarrollo obliga a un manejo rústico de dichos procesos. Por otro lado, la infraestructura, es decir el conjunto de terminales que se alojan en los servidores, que no son monitoreadas de manera adecuada, pueden presentar fallas a largo plazo, o desconexiones que priven a todo el sistema de sus funcionalidades.

Las consecuencias pueden no manifestarse en un corto plazo, puesto que los sistemas son gestionados con el software propietario de cada servidor, pero no es apto para gestionar incidentes propios de cada host. Manejar procesos, conexiones a operadoras y bases de datos, pueden facilitar que los servicios sufran caídas, que deben ser solventados en el menor tiempo posible, por ello la necesidad de un sistema de monitoreo específico. Descrita en (Aquino Quiñonez, 2017)

Por el crecimiento de la empresa y las nuevas plataformas que se están desarrollando, es necesario implementar una plataforma de control y monitoreo, para evitar tiempos innecesarios de resolución de problemas que se pueden automatizar, y el escaso monitoreo de terminales. Además, el entorno de código abierto permite herramientas robustas, que se pueden acoplar al esquema de la empresa Message Plus. Lo que permite el desarrollo de una plataforma que cumpla las expectativas de la empresa, descritas en el desarrollo de este proyecto.

Problema

La empresa Message Plus requiere una plataforma de monitoreo y control de sus terminales y procesos principales, puesto que inconvenientes en estos son solventados a medida que son descubiertos por el personal de soporte, lo que causa un tiempo de respuesta no óptimo en el servicio. Una plataforma de las características mencionadas puede reducir el tiempo de atención a los requerimientos, puesto que automatiza las acciones que sean recurrentes, ya que algunos procesos se pueden manejar de forma autónoma por la plataforma, y ser atendidos con agilidad con una notificación oportuna de la misma al personal de soporte.

Justificación

El proyecto implementa la plataforma de manejo y control de procesos, que gestiona la infraestructura lógica de la empresa, y permite formar un sistema de alertas en base al grado de criticidad de los procesos que se manejan en estos ambientes.

El proyecto es aplicado en el ambiente de producción de la empresa Message Plus, con un grupo de cerca de 80 clientes que usan los servicios que la empresa ofrece, (se puede revisar en la página oficial, (Message Plus S.A., 2019)). Algunos de estos procesos y terminales ahora serán monitoreados y controlados por una plataforma autónoma siempre que sea posible, ofreciendo un sistema capaz de manejar la infraestructura lógica y los servicios en tiempo real.

Objetivos

Objetivo general.

Diseñar e implementar un sistema centralizado de monitoreo, supervisión y control automático de servidores y servicios en entornos virtuales de la empresa Message Plus basado en herramientas de código abierto.

Objetivos específicos.

Analizar la infraestructura de la empresa, con el desglose de las actividades principales de cada servidor, y los procesos que se ejecutan en ellos.

Establecer los principales servicios para ser monitorizados, con un sistema jerárquico y la posibilidad de ser controlados de manera automática.

Comparar las herramientas de código libre, con sus principales características en entornos similares a la arquitectura analizada para ser implementadas en la plataforma de control y monitoreo.

Implementar la plataforma de control y monitoreo de servidores y servicios, con las herramientas de código libre seleccionadas, en una interfaz web, montada sobre un cliente alojado en los servidores de la empresa.

Implementar la aplicación móvil de monitoreo de servidores y servicios para entornos con sistema Android, basada en la implementación web.

Evaluar la plataforma de control y monitoreo de servidores y servicios con pruebas afines a la arquitectura de la empresa.

Evaluar económicamente la solución planteada, comparando con entornos licenciados, en arquitecturas similares.

Metodología

El proyecto responde a la metodología Scrum, siendo un trabajo incremental de desarrollo por bloques, con un resultado respaldado con pruebas, como se explica en (Daniel, 2019). Se plantean una serie de actividades que llevan a la formación de la plataforma con una serie de módulos, que en conjunto presentan el sistema terminado.

Está dirigido para el personal de sistemas de la empresa Message Plus, como una plataforma de control y manejo de la infraestructura y servicios que ofrece la empresa.

Se ubicará en un host dentro de la arquitectura de la empresa, en un plazo de 4 a 6 meses, donde se hará toma de datos, desarrollo y pruebas para entregar un sitio web completo que tenga todo lo requerido.

Está destinado a ser una herramienta de control y manejo capaz de automatizar tareas, y servir como referencia a otras empresas que deseen plantear un sistema a fin.

Capítulo II

2.1. Marco teórico

2.1.1. Datacenter.

El centro de procesamiento de datos es un entorno de amplio espacio, donde se alojan equipos electrónicos que gestionan la información de una organización, estos pueden dedicarse también a prestar este servicio a más organizaciones, gestionando equipos de hardware, y velando por las condiciones adecuadas para su buen funcionamiento. Como se define en (Alvarez, 2015), los principales objetivos que busca un Datacenter son.

Tabla 1. Objetivos Datacenter

Suministro eléctrico continuo	**Este es el primer apartado principal de un datacenter, las condiciones de todo tipo de infraestructura con equipos electrónicos, requiere de un suministro estable que garantice la energía, para precautelar los equipos y dar un servicio sin interrupciones.**
Conexión a internet	El segundo apartado con mayor relevancia, es la conexión de los equipos a la red, ya que muchos de los servicios que se alojan en un centro de procesamiento de datos son accedidos desde lugares muy distantes, por ellos son cableados y poseen una conexión redundante, que garantiza la disponibilidad.
Controles de	La información es el activo más importante de una

seguridad	organización, y el cuidar los hosts donde se aloja es una prioridad. Se manejan sistemas de control físico, como cámaras y puertas de acceso controladas, pero es aún más necesario el control de seguridad del software, los sistemas poseen firewalls de control, que previenen las intrusiones no autorizadas, y sistemas que independizan los servicios de empresa a empresa, aun así, mucha de la responsabilidad de este control recae en la organización que usa el hardware.
Control de climatización	Para un correcto funcionamiento de equipos electrónicos, existen condiciones favorables, las cuales son emuladas en un datacenter, así no solo se logra el máximo desempeño del hardware, también la prevención de problemas. La temperatura oscila entre los 16 y 25 grados.

Fuente: (Alvarez, 2015)

2.1.2. Administración de red.

Según el sitio oficial de Telconet (2020) la administración de una red se basa en la organización, gestión y control de los eventos que ocurren en ella. Incluye el mantenimiento y gestión del hardware como. Routers, Switches, Firewalls entre otros. Además de actividades como la configuración de direcciones, habilitación de puertos, tablas de ruteo, autenticación y gestión de servicios.

En toda arquitectura de red, existe un administrador, que gestiona la realización de las actividades, esto se puede dividir en estas áreas.

Tabla 2. Actividades de la administración

a. Atención al cliente
b. Diseño de la arquitectura de red, en base a requerimientos y con un grado de escalabilidad alto.
c. Sostenible ante cambios
d. Gestión y mantenimiento
e. Tolerante a fallos
f. Red auditable
g. Plan de escalabilidad a corto y largo plazo

Fuente: (Alvarez, 2015)

2.1.3. Gestión del software en la red.

La red la componen un conjunto de elementos de hardware que se conectan entre sí, como se muestra en la figura 1. Esta red dispone de un conjunto de terminales y aplicaciones de software, que conectan lógicamente estos componentes, ahora se revisa como funciona esta estructura y que elementos y sistemas intervienen en su gestión y configuración

Los elementos lógicos de la red, son terminales que gestionan el mantenimiento de su sistema, enviando alertas en caso de un fallo, y analizando esas situaciones para ejecutar acciones que salvaguarden la información, estos mensajes pueden ser gestionados y automatizados con un software especializado, o tratados de manera manual en cada caso, todo definido por los administradores de red. Como se describe también en (Aquino Quiñonez, 2017).

Tabla 3. Roles de la organización TI

Administrador	**Los administradores de red, son los encargados de la revisión de las notificaciones generadas por esos terminales. La verificación se puede revisar manualmente o establecer acciones que automaticen el proceso, esto con agentes que revisen el estado de las terminales, pero para un correcto funcionamiento del sistema, todas las notificaciones deben ser atendidas.**
Agente	Los agentes son aplicaciones que se alojan en las terminales para la gestión de las notificaciones que estas producen, se instalan de acuerdo al software que se quiere emplear. Estos agentes agrupan una gran cantidad de información y la envían al servidor del software, en donde se organiza y se presenta al administrador de una manera efectiva y clara. Este intercambio de información se realiza mediante el uso de protocolos de red, como el SNMP (Protocolo de administración de red).

Fuente: (Aquino Quiñonez, 2017)

Organización de red

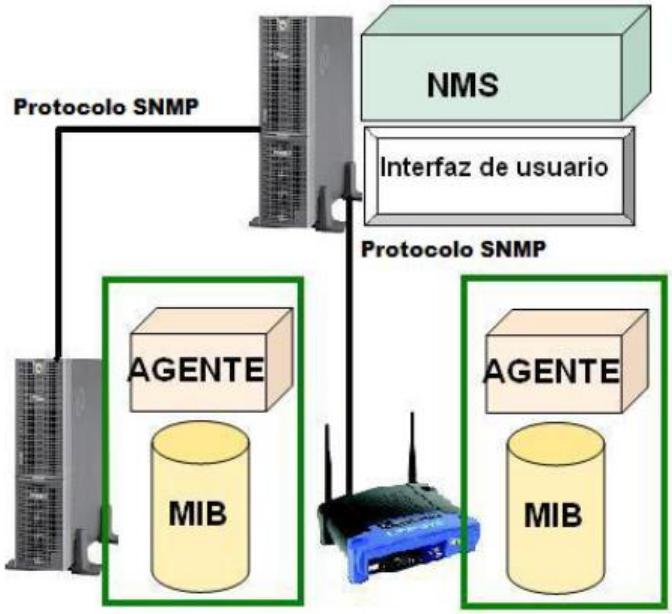


Figura 1. Estructura de la organización de red

Tomado de: (Sergio, 2020)

2.1.4. SNMP (Protocolo simple de gestión de red).

Según (Aquino Quiñonez, 2017) SNMP es un protocolo de gestión de red por medio del protocolo TCP/IP, con el modelo de servidor – agente. Se basa en la monitorización de un grupo de agentes en un host de administración, siendo escalable a muchos tipos de configuración y tecnologías de distintos fabricantes.

Este protocolo proporciona un medio estandarizado para el control de los elementos de la red, siendo simple y eficaz, permite el intercambio de mensajes del host de administración y los agentes sin un impacto fuerte en el rendimiento de la red. Esta comunicación se hace por paquetes UDP, por los puertos 161 y 162. SNMP busca un conjunto de objetivos para la monitorización, que se definen a continuación.

Tabla 4. Objetivos de SNMP

Eficiencia en el uso de recursos.
Monitorización y control de agentes para una respuesta más corta ante problemas de recursos.
Automatización de eventos ya conocidos.
Seguimiento de los cambios y acciones en la red.
Control de actualizaciones.
Escalabilidad controlada.

Fuente: (Aquino Quiñonez, 2017)

El protocolo SNMP, presenta las siguientes características principales.

Tabla 5. Características de SNMP

Trabaja en la capa de aplicación del modelo TCP/IP.
Tiene la arquitectura cliente – servidor, donde un host es el administrador que intercambia mensajes con los agentes instalados en otros hosts.
Utiliza un paquete de 64 Kb, que usan el protocolo UDP.
Presenta un sistema de continuo monitoreo con envío y recepción de mensajes en la red, para garantizar la confiabilidad e la información.
Es simple para el manejo de sistemas de monitorización, pero sin un gestor bien definido se puede convertir en un sistema de difícil ampliación.

Fuente: (Aquino Quiñonez, 2017)

2.2. Software de código abierto

El software de código abierto como se detalla en (GitHub, 2020), es todo aquel que se puede tomar de forma libre, modificar y publicar nuevamente bajo cambios

específicos, sin ningún fin de lucro. Aunque es un término inicializado con el auge de las computadoras, el concepto ha cambiado, ya que no solo son programas específicos, sino entornos completos que se pueden acoplar a tareas específicas.

El código abierto presenta la fuente de dicho software para que sea utilizada, evaluada y mejorada por nuevos usuarios, ya que la licencia es de acceso libre, y pretende mejorar las características de los programas o corregir defectos que no permiten un correcto funcionamiento.

2.2.1. Licencia de software de código abierto contra software propietario.

Quizá el software más relevante es el propietario, en donde solo la persona o grupo de personas que desarrollaron la fuente de un programa pueden mantener un control sobre dicha fuente, las modificaciones o mejoras están disponibles bajo la compra de una licencia que incluye estas adaptaciones o bajo nuevos parches que pueden representar un valor adicional, sin permitir un control sobre las mismas. (Oscar, 2017)

Ahora una licencia de código abierto es completamente diferente a la propietaria, esta permite al usuario o desarrollador tener todo el control sobre el software, desde la fuente hasta su modificación y distribución. El acceso de manera local es completamente libre, los cambios en el software pueden hacerse en completa libertad, pero cuando se quiere volver a publicar ese software con cambios, la licencia obliga a que dicho código sea liberado sin ningún costo, con el único fin de mejorar la colaboración de la comunidad.

2.2.2. Porque software de código abierto, y no el propietario.

Existen una gran variedad de razones para preferir el uso de herramientas de código abierto, a continuación, se explican las indicadas por (Alvarez, 2015).

Tabla 6. Beneficio de herramientas de código abierto

Control	**El control sobre el software es casi completo, mientras su fin no sea cambiado, el desarrollador es libre de realizar cualquier modificación si es que algún apartado del software no se acopla a la situación del proyecto.**
Formación	La formación que permite la manipulación directa del código permite la adquisición de habilidades mientras se realiza un proyecto, además de promulgar una comunidad de ayuda, trabajando bajo los mismos parámetros.
Seguridad	La seguridad es un apartado importante, ya que, al haber un número grande de desarrolladores bajo un proyecto, es posible que un fallo sea detectado con mayor rapidez, y que la solución depende de la misma comunidad que le da uso, lo que brinda más estabilidad y agilidad en actualizaciones de corrección.
Estabilidad	La estabilidad que provee un software de código abierto, es uno de los ítems más valorados por la comunidad, al ser de acceso libre y que muchas personas disfruten de sus otros beneficios, permiten que el software no se deteriore, y quede discontinuado, al no pertenecer a un grupo de propietarios, este estará disponible en línea hasta que el último desarrollador deje de usarlo, un hecho poco probable.

Fuente: (Carlos Velasco, 2017)

2.3.Herramientas de monitoreo de código abierto

El poder monitorear un entorno permite garantizar que un sistema funcionará en perfecto estado todo el tiempo. Además de prevenir posibles fallos antes de que se presenten, lo que puede ahorrar recursos y tiempo del equipo.

Existe una gran cantidad de software libre para la monitorización de un entorno, encargado de verificar la infraestructura y avisar de cualquier situación anómala, ahora se analizará un grupo de los mejor valorados por la comunidad (GitHub, 2020).

2.3.1. Nagios.

Es uno de los líderes en el monitoreo de entornos, capaz de dar seguimiento a casi cualquier componente de una red, como protocolos, aplicaciones, servidores web, entro otros. Con un consumo mínimo de recursos por su monitoreo en Core 4, y adaptabilidad a la integración con otros softwares, lo hace muy robusto en soluciones web empresariales, como se explica en (Zamora, 2013).

Entre sus mejores características se puede encontrar la vista centralizada de toda la infraestructura de TI como se indica en la figura 2, además de un gestor de reinicio de aplicaciones ante fallos del sistema, permite un acceso de múltiples usuarios y una segregación de vistas dependiendo del usuario que accede al servicio.

Nagios

Figura 2 Panel principal de Nagios

Tomado de: (Kumar, 2020)

2.3.2. Cacti.

Es una herramienta de monitoreo de red muy completa como se detalla en (Wong, 2012), se forma bajo SNMP y presenta una gran cantidad de grafos estadísticos fáciles de interpretar como se muestra en la figura 3. Entre sus principales características está el establecer partes de los gráficos ilimitados con fuentes de datos directos del repositorio de Cacti, también el soporte de autocompletado de gráficos. El permitir archivos RRD, deja a Cacti trabajar con muchas fuentes a la vez en línea o el uso de archivos locales para su gestión de sistema, junto con esto el control de usuarios y los script de inteligencia artificial, para mejorar la presentación de datos al usuario final.

Cacti

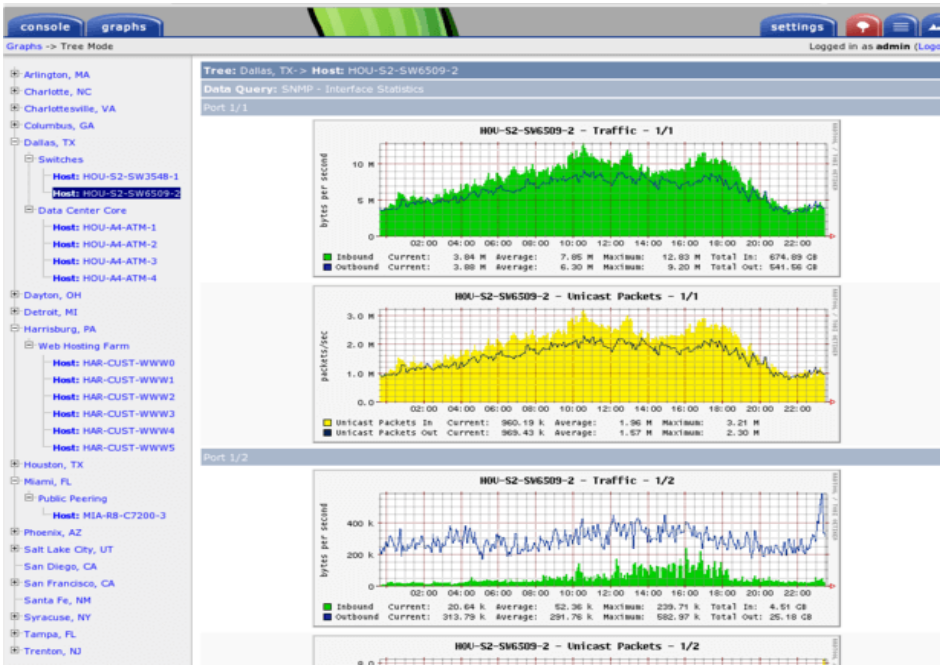


Figura 3. Panel principal de Cacti

Tomado de: (Kumar, 2020)

2.3.3. OpenNMS.

Es una solución que permite la creación de un entorno de monitoreo de infraestructura de TI, describe la topología de red de capa 2, basada sobre una estructura de eventos. Con una interfaz gráfica accesible e intuitiva como se muestra en la figura 4, capaz de correr en Docker cada uno de sus apartados. A pesar de ser diseñado para entornos Linux, su desarrolla ya permite integración con otros sistemas operativos como se explica en (OpenNMS, 2020).

OpenNMS

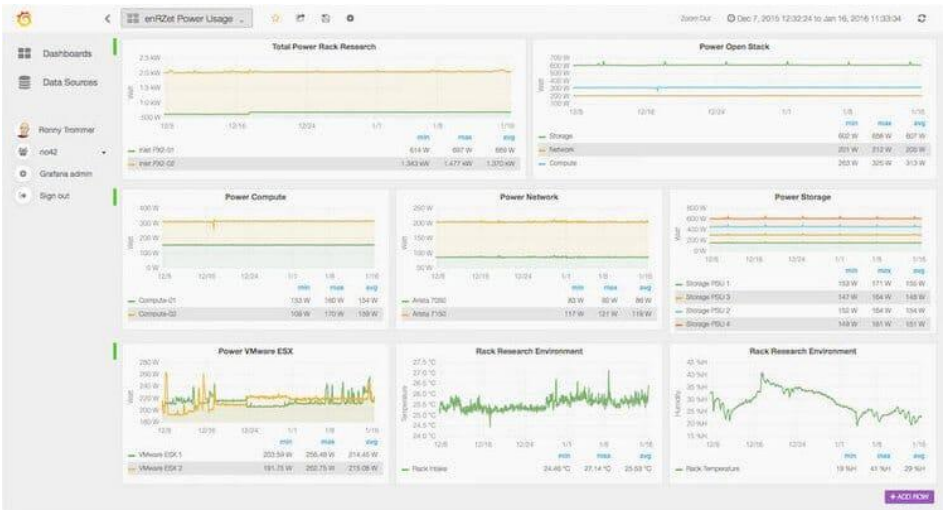


Figura 4.Panel principal OpenNMS

Tomado de: (Daniel, 2019)

2.3.4. Zabbix.

Software de monitoreo de red como se muestra en la figura 5, que puede registrar y hacer seguimiento de un grupo de sistemas, como servidores, servicios, host y recursos de la red. Permite un alto desempeño en la gestión de recursos, la centralización de recursos, y la implementación de un ambiente gráfico que ayude a los operarios en el manejo de incidentes, como lo describe (Zamora, 2013).

Zabbix

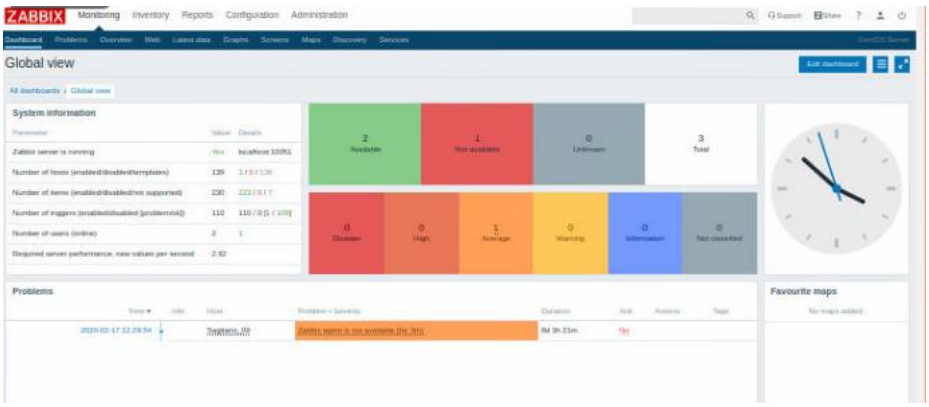


Figura 5. Panel principal Zabbix

Tomado de: Autor

2.3.5. Monit.

Software para la administración y monitoreo de procesos, directorios, sistemas operativos para sistemas basados en Unix. Especializado en el mantenimiento y cuidado automático de procesos, con acciones programadas que pueden ser ejecutados por una secuencia de ocurrencias también especificadas, como el control de un proceso que debe estar corriendo en todo momento, o el manejo de recursos de un sistema, como se muestra en la figura 6.

Monit

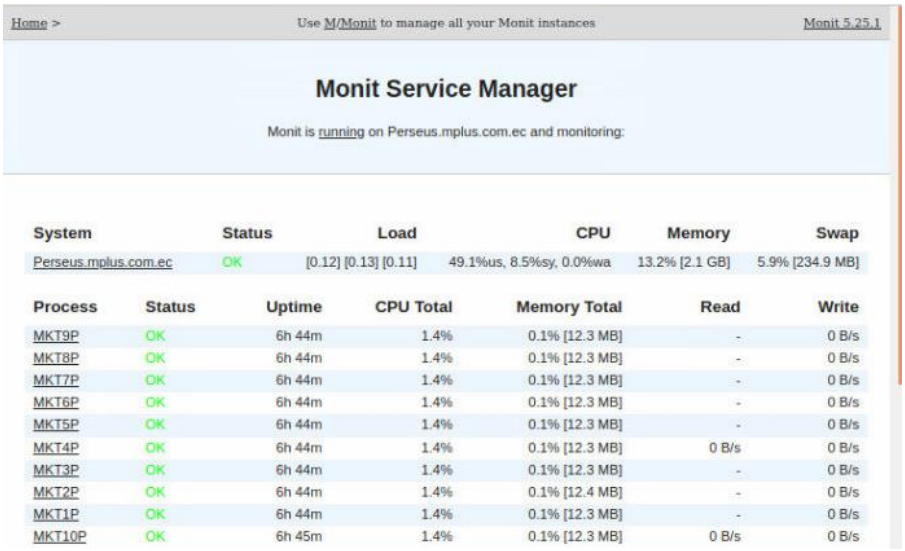


Figura 6. Panel principal Monit

Tomado de: Autor

2.3.6. Munin.

Herramienta de monitoreo de red, que analiza la utilización de los recursos de la red, y la tendencia para una mejor optimización como se muestra en la figura 7. Permite la monitorización de procesos de servidores, aplicaciones y diversos ámbitos

de un host, junto con la mejora del consumo de recursos. Usa la arquitectura de maestro – nodo, que permite la centralización de la infraestructura.

Munin

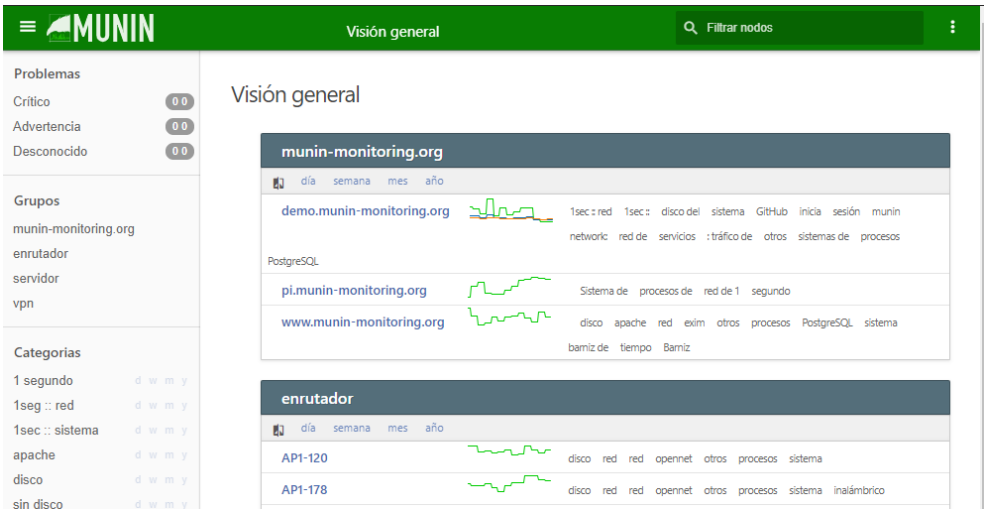


Figura 7. Panel principal demo Munin

Tomado de: (MUNIN, 2020)

2.4. Comparación de Software libre

Una vez descritas, en el apartado anterior, las mejores opciones que se pueden acceder de manera libre, se procede mediante cuadros comparativos a escoger la mejor alternativa para la solución que se desea implementar, los aspectos considerados en este apartado, se acoplan a esta solución, pero pueden ser manejados bajo parámetros iguales en ambientes diferentes, aunque nuevas soluciones pueden tomar nuevos aspectos y afectar a nuevas elecciones de software.

Además, se hace la diferenciación entre software libre para el manejo de recursos de hardware y control de sistemas y el software para el control de procesos.

La comparación usará una escala de 0 a 10, siendo 0 el valor que describa el menor desempeño y 10 el mejor desempeño en el ámbito analizado.

2.4.1. Software de infraestructura.

Tabla 7. Comparación de software de infraestructura

Aspecto/Software	Nagios	Cacti	OpenNMS	Zabbix
Interfaz sencilla, vista centralizada de toda la infraestructura.	8	6	7	9
Control de usuarios, acceso.	7	6	6	8
Manejo sencillo de expansiones en la infraestructura.	9	6	8	9
Encriptación de conexiones, dentro de la red con cada host.	8	7	7	9
Integración con herramientas externas, que faciliten la gestión.	9	8	8	9
Número de funciones disponibles.	9	8	9	9
Comunidad de desarrollo y soporte para el software.	9	8	9	9

Elaborado por: Autor

2.4.2. Software de control de procesos.

Tabla 8. Comparación de software de control de procesos

Aspecto/Software	Monit	Munin
Implementación sencilla, interfaz simple e intuitiva.	8	9
Monitoreo y administración sencilla de sistemas de información.	9	7

Monitoreo de recurso de red.	5	9
Manejo de tareas de mantenimiento, reparación y ejecución automáticas.	9	6
Supervisión del rendimiento de los recursos, que busque posibles problemas en el futuro.	7	9
Comunidad de desarrollo y soporte para el software.	9	7

Elaborado por: Autor

2.4.3. Resultados de la comparación.

En cuanto al software de infraestructura se eligió Zabbix por la interfaz sencilla que brinda, con un conjunto de gráficas y notificaciones intuitivas, que hacen el monitoreo de servidores más práctico. Además de su fácil integración con VMWare, software que maneja la empresa para la gestión de sus MV. También por la posibilidad de conectar a los clientes por claves pre-compartidas, y la encriptación de las conexiones. En cuanto al manejo de notificaciones, presenta una integración completa con herramientas como PushOver, y PushBullet, que facilitan el manejo de alertas independientemente de la plataforma donde se haga el monitoreo.

Para el software de control de procesos, se decidió usar a Monit, por que las características más fuertes de este software se acoplan mejor al sistema que se desea implementar, debido a que permite un monitoreo y administración de sistemas de información mucho más completo, con el uso de los script que solo respetando su estructura base pueden acoplarse a cualquier proceso, independiente del lenguaje de programación. Adicional permite la reparación, y ejecución automática de acciones

en situaciones que presenten un problema para el sistema. Con una interfaz simple que deje ver el estado y funcionamiento de cada proceso.

2.5. Monitoreo de los elementos red

Este apartado se basa en el control, verificación y monitoreo de las diferentes terminales que componen la red, además de los servicios que esta puede prestar. Este control se realiza por medio del uso de un software dedicado a este fin, que se encarga de verificar el estado de sus agentes, y reuniendo toda la información en una sola terminal de administración. El fin principal de esta terminal, es encontrar y notificar incidentes a la brevedad, así como elementos que presentan algún daño o signifiquen una disminución en el rendimiento de la red.

2.5.1. Bases del monitoreo de elementos de red.

El monitoreo de red se define según (Solis, 2014), en cuatro fases elementales, que abarcan los principales aspectos, que son los siguientes.

Tabla 9. Fases del monitoreo de elementos de red

Administración	**Establecer la información que la terminal de administración maneja; es decir definir qué información es la que los agentes deben establecer para que la administración monitorea el estado de los elementos de red.**
Información	Medio de obtención de la información; es como la administración y los agentes se comunican dentro de la red, los puertos que deben usar y si se establece, un método de encriptación

Políticas	Definición de políticas; son las normas que se establecen en la terminal de administración para aplicar a los agentes.
Procesamiento	Procesamiento; es como se trata la información que generan los agentes, y como esta se presenta en una interfaz a los usuarios que les compete.

Fuente: (Solis, 2014)

2.5.2. Monitoreo de terminales.

Es la verificación del estado funcional de las máquinas virtuales, aquellas que son creadas para alojar servicios, procesos, bases de datos entre otras aplicaciones de software que requieren de un sistema operativo para funcionar. Pero este monitoreo se especializa en el seguimiento de las características de la terminal, como la ocupación del disco asignado, el consumo de la memoria, la carga que se le da al procesador o el tamaño de una base de datos, información que prevé la disponibilidad de la información y garantizan la continuidad del negocio, como se describe en (Wilman, 2014)

2.5.3. Monitoreo de servicios.

El control de procesos en una organización que ofrece servicios con disponibilidad 24/7 es imprescindible, pues la base de su sistema empresarial es el ofrecer su sistema en cualquier momento y bajo cualquier situación, pero el control de esos procesos que se ejecutan en una terminal, se alejan de la verificación del estado de la misma. El control y monitoreo ahora se centra en la ejecución de aplicaciones, que realizan una tarea, aquella que deben ser revisadas constantemente

y mecanizadas, para aumentar el rendimiento de todo el sistema y acortar la tolerancia a fallos que se puedan automatizar.

2.6. Recursos de software de las terminales

Los recursos son el conjunto de elementos de los que se dispone en la red, en este caso, aquellos que forman la estructura a la que se va a implementar el software de control y monitoreo. Es importante contextualizar todas estas definiciones para establecer procedimientos acordes a la infraestructura lógica que se tiene. Además, en un sistema informático, el versionamiento ayuda en aspectos como la seguridad, el rendimiento y la calidad de los servicios que se ejecutan en las terminales.

2.6.1. Centos.

Según (CentOS Org, 2020), este es un software libre que se basa en los aportes de la comunidad, para lograr un entorno robusto y completo sobre la GNU Linux. La plataforma integra una gran cantidad de aplicaciones e implementaciones basadas en código abierto, que permite su uso y modificación para el acoplamiento a distintas necesidades. Las ventajas de CentOs son de utilidad para el manejo de servidores y bases de datos, donde la seguridad es importante, y el rendimiento frente a situaciones complejas impera. Existe una gran cantidad de versiones de este sistema operativo, que, aunque se base en el mismo core, difiere el manejo de la terminal, de acuerdo a los paquetes que se desea utilizar, por lo que es importante revisar si la codificación es correcta para la versión que se maneja en la implementación.

2.6.2. MariaDB.

Según (MariaDB, 2020), este es un administrador de BDD con un conjunto de funcionalidades especializadas en el manejo de información. Es de acceso libre y es desarrollado por ingenieros creadores de MySQL, pero sin perder el objetivo principal de un desarrollo libre. De gran presencia en ambientes Linux, basa su funcionamiento en el rendimiento, la estabilidad, y la apertura, es decir la fácil adaptación a entornos distintos con sistemas y aplicaciones propietarias.

2.6.3. Postgres.

Según se define en (PostgresSQL, 2020), este es un sistema de bases de datos relacionales de código abierto, con una comunidad activa empeñada en su desarrollo, además de una sólida gama de funcionalidades que garantizan su rendimiento y robustez.

La documentación es amplia y el un gestor que se abre paso a ser de los más consumidos en plataformas Linux, por su arquitectura, fiabilidad, integración a sistemas, capacidad de crecimiento y filosofía abierta. Además, su implementación es simple y no requiere de una inversión grande.

Hoy en día el usar postgres garantiza la protección de datos y el desarrollo de entornos tolerantes a fallos, que pueden implementar acciones de automatización de eventos. Posee una amplia compatibilidad con otros gestores para migraciones controladas, y adaptación de funcionalidades, con soporte para múltiples tipos de datos.

2.6.4. PHP.

Como se define en (PHP, 2020), este es un lenguaje de programación desarrollado y especializado en la programación web. Siendo rápido, extensible y pragmático, se ha abierto campo en la web. Con un gran apoyo de la comunidad, lanza periódicamente nuevas versiones para la mejora de la seguridad, y la ampliación de funcionalidades.

2.6.5. VMWare.

Software licenciado que permite la creación y administración de entornos virtuales, para garantizar la disponibilidad de los mismos, da el soporte necesario para su gestión y desarrollo de actividades de usuarios finales, además define los protocolos de actualización y seguridad necesarios para que el sistema sea sostenible. Su objetivo es priorizar la movilidad de sus usuarios, con el modelo cliente – servidor conectados por una nube.

Capítulo III

3.1.Marco metodológico

El proyecto responde a la metodología Scrum, siendo un trabajo incremental de desarrollo por bloques, con un resultado respaldado con pruebas, como se explica en (Daniel, 2019). Se plantean una serie de actividades que llevan a la formación de la plataforma con una serie de módulos, que en conjunto presentan el sistema terminado.

Está dirigido para el personal de sistemas de la empresa Message Plus, como una plataforma de control y manejo de la infraestructura y servicios que ofrece la empresa.

Se ubicará en un host dentro de la arquitectura de la empresa, en un plazo de 4 a 6 meses, donde se hará toma de datos, desarrollo y pruebas para entregar un sitio web completo que tenga todo lo requerido.

Está destinado a ser una herramienta de control y manejo capaz de automatizar tareas, y servir como referencia a otras empresas que deseen plantear un sistema a fin.

La investigación será bibliográfica, describiendo cada uno de las partes que contempla el trabajo, hasta llegar a las herramientas específicas del proyecto, tomando en cuenta que mucho del proyecto se basa en software libre, se citan sitios de este tipo, como Github o GitLab, además de foros y páginas oficiales de intercambio de información.

La investigación de campo será solo para estar al tanto del estado de la infraestructura y lo que funciona en cada uno de los clientes. Como se detalla en (Aquino Quiñonez, 2017)

Se implementará una aplicación móvil en base a la plataforma web, que permita el control y monitoreo de los servidores y servicios, haciendo de la plataforma una herramienta portable, con funcionalidades básicas, pero de gran utilidad en momentos de presión.

El siguiente es el eje estructural que el proyecto cumple en su desarrollo, describiendo de qué manera se realiza cada apartado y los ejes que lo conforman, con las herramientas y actores involucrados.

Tabla 10. Eje estructural del proyecto

Eje	Investigación	Metodología	Técnica	Instrumento	Fuente
Analizar la infraestructura de la empresa, con el desglose de las actividades principales de cada servidor, y los procesos que se ejecutan en ellos.	Experimental	Cualitativa Cuantitativa Scrum	Entrevista Abierta Análisis de contenido Iteraciones con la infraestructura	Cuestionario Archivos MessagePlus Plataforma VMWare	Gerente TI Departamento TI Servidores
Establecer los principales servicios para ser monitorizados, con un sistema jerárquico y la posibilidad de ser controlados de manera automática.	Experimental	Cualitativa Scrum	Entrevista Abierta Iteraciones con los host	Cuestionario Putty	Gerente TI Servidores
Comparar las herramientas de código libre, con sus principales características en entornos similares a la arquitectura analizada para ser implementadas en la plataforma	Experimental	Cualitativa	Análisis de documentos	Archivos	Tesis, artículos, Páginas oficiales

de control y monitoreo.					
Implementar la plataforma de control y monitoreo de servidores y servicios, con las herramientas de código libre seleccionadas, en una interfaz web, montada sobre un cliente alojado en los servidores de la empresa.	Experimental	Scrum	Iteraciones de desarrollo	Herramientas de código abierto seleccionadas, PC.	Páginas oficiales, Guías, Tesis y artículos
Implementar la aplicación móvil de monitoreo de servidores y servicios para entornos con sistema Android, basada en la implementación web.	Experimental	Scrum	Iteraciones de desarrollo	Android Studio, xamarin	Plataforma implementada en la web
Evaluar la plataforma de control y monitoreo de servidores y servicios con pruebas afines a la arquitectura de la empresa.	Experimental	Scrum	Iteraciones de pruebas	Ambiente de pruebas de MessagePlus	Documento de QA de MessagePlus
Evaluar económicamente la solución planteada, comparando con entornos licenciados, en arquitecturas similares.	Experimental	Cualitativa	Análisis de documentos	Archivos	Tesis, artículos, Páginas oficiales

Elaborado por: Autor

3.2.Infraestructura empresarial

La empresa MessagePlus posee una infraestructura física compuesta por 3 servidores HPE, estos equipos se alojan en un rack en la organización Telconet, donde se contrata el servicio de hosting, esto se refiere a que esta empresa se comprometo a garantizar el funcionamiento continuo de los servidores, administrar sus conexiones y configuraciones de los switches y el firewall [1]. Para el desglose de la organización lógica se entabló una entrevista con el Gerente de TI, el señor

[1] Información tomada de la entrevista con el Gerente de TI, que se puede revisar en anexos.

Patricio Cuvi, para conocer cómo se organiza la empresa y cuál es la arquitectura que se plantea monitorear. En el siguiente apartado se describen los aspectos relevantes sobre esta reunión, indicando que algunos de los host que se presentan son de carácter confidencial por lo que su identificación ha sido modificada, sin alterar la base estructural en sí.

3.2.1. Estructura de la entrevista.

La entrevista tiene el propósito de conocer cuál es la estructura lógica de la empresa, documentar en diagramas sencillos las terminales que se desea monitorear, e identificar los recursos de red que interviene en los servicios que presta la empresa. Esta reunión se realizó con un carácter abierto, ya que las nociones necesarias para el planteamiento de la solución, se distinguen a aspectos básicos de la estructura lógica y al conocimiento de existencia de los procesos que se automatizarán.

3.2.2. Análisis de la entrevista.

Con la realización de la entrevista, se han obtenido los elementos principales de la infraestructura lógica de la empresa, y se presenta a continuación los apartados más relevantes para la realización de la plataforma.

- La empresa MessagePlus se dedica a brindar servicios de mensajería masiva, uso de IVR y mailing, además de análisis en el apartado de marketing y publicidad. Por lo que tiene conexiones con las distintas operadoras del país para el despacho de estos servicios en conformidad a la ley de telecomunicaciones vigente en el Ecuador.

- Los servicios de mayor importancia en la empresa son los encargados de la conexión con las bases de datos, estos se actualizan en tiempo real, y son los encargados se registrar todo el tráfico que maneja la empresa, los cuales se usan para facturación.
- Después los servicios de conexión a las operadoras por donde se realizan los despachos de SMS o IVR, que se asignan a cada cliente en el plan contratado, estos servicios se monitorean por la consola de la terminal Linux donde se alojan, y se detallan en el cuadro siguiente.

Tabla 11. Servicios a disposición

Operador	Servicio/Conexiones abiertas	Descripción
Conecel	Conexión Rx (1)	Encargada de la recepción de datos desde el usuario.
Conecel	Conexión Tx (6)	Encargada del despacho de datos informativos hacia los usuarios para Conecel.
Conecel	Conexión Marketing (4)	Encargada del despacho de datos referentes a marketing hacia los usuarios para Conecel
Otecel	Conexión Rx (1)	Encargada de la recepción de datos desde el usuario.
Otecel	Conexión Tx (4)	Encargada del despacho de datos informativos hacia los usuarios para Otecel.
Otecel	Conexión Marketing (2)	Encargada del despacho de datos referentes a marketing hacia los

		usuarios para Otecel.
Corporación Nacional de Telecomunicaciones	Conexión Rx (1)	Encargada de la recepción de datos desde el usuario.
Corporación Nacional de Telecomunicaciones	Conexión Tx (3)	Encargada del despacho de datos informativos hacia los usuarios para CNT.
Corporación Nacional de Telecomunicaciones	Conexión Marketing (2)	Encargada del despacho de datos referentes a marketing hacia los usuarios para CNT.

Elaborado por: Autor

- Los tres servidores físicos se encuentran en red, pero no están clusterizados, no comparten recursos y son independientes entre sí, la conectividad está a cargo de 2 switches, un esclavo y un master.
- Los sistemas que operan en la red son íntegramente CentOS, lo que difiere entre algunas de las terminales es la versión, que pueden ser. CentOS 6.1, CentOS 6.9 y CentOS 7, por lo que se debe tener en cuenta esto al momento de realizar cualquier instalación.
- El personal de soporte de TI de la empresa realiza acciones en los servidores por medio del entorno de virtualización VMWare, que posee una terminal en los servidores con una IP pública para que se pueda acceder desde lugares remotos. Pero no existe un sistema de control de incidentes especializado, en donde el departamento de soporte sea alertado de incidentes, o automatice acciones en caso de ser necesario.

- El conjunto de procesos que rigen las conexiones a la base de datos y a las operadoras no son controlados, ni poseen un sistema de monitoreo y automatización, solo se reinician a intervalos de tiempo definidos por el crontab de Linux, para garantizar la disponibilidad de los mismos.

3.2.3. Comprensión de requerimientos.

Una vez recopilada la información, se plantea los requerimientos que la plataforma debe satisfacer, y que son factibles implementar en un entorno implementado en base a las herramientas de código abierto que se seleccionaron.

La empresa desea implementar una plataforma que permita el control y monitoreo de sus terminales y de sus principales servicios, a la vez que se automatice la gestión de algunos procesos que no requieran de la supervisión de un operario para su corrección, pero que permita evidenciar las acciones que realiza.

Los resultados esperados son.

- Plataforma de monitoreo de hosts.
- Plataforma de control y monitoreo de los servicios principales.
- Adaptación de la plataforma a un entorno móvil.

3.2.4. Arquitectura de hosts

En base a la entrevista, se presenta el esquema de hosts que se tienen en la empresa actualmente, estos hosts son los que se monitorizaran en la plataforma. Su definición en la imagen es referencial y se establece con el fin de la familiarización con el escenario, se plantea definir este concepto en la plataforma. Esta arquitectura se detalla en la figura 8.

Arquitectura de hosts

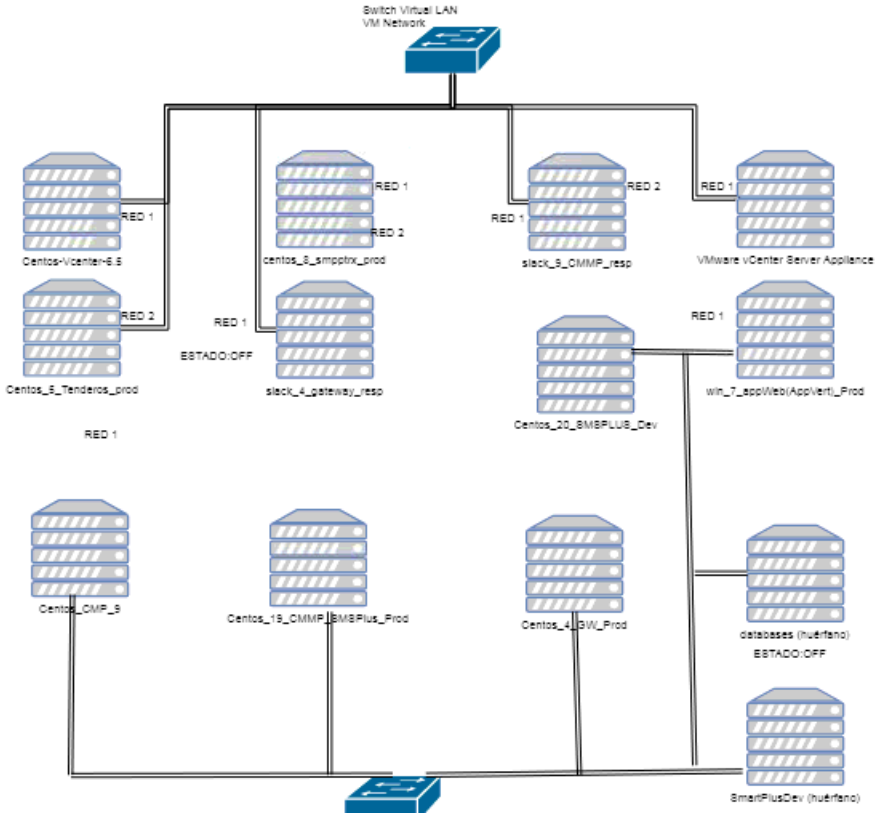


Figura 8. Arquitectura de hosts

Elaborado por: Autor

3.3.Servicios para monitorización

Una vez determinados los procesos que la empresa usa para la gestión de sus servicios, se establece el sistema jerárquico que se implementará para el tipo de notificación al departamento de TI. Este sistema se evalúa en base a la importancia de cada proceso, el impacto que tiene su tiempo de inactividad y la disponibilidad ante situaciones de emergencia.

3.3.1. Jerarquía de servicios (criticidad).

Existen por cada operador actualmente en el país, (Conecel, Otecel y Corporación Nacional de Telecomunicaciones), un grupo de 3 distintas conexiones, que se realizan por procesos que la empresa levanta con esos operadores, en distinto número, por razones comerciales y de demanda. Estas conexiones tienen un grado de importancia distinto entre sí por el servicio que prestan a sus usuarios, pero que son iguales en todas las operadoras, por ello se define la siguiente estructura en base a la importancia de esos procesos.

Tabla 12. Criticidad de los servicios

Servicio/Conexiones abiertas	Descripción	Grado de criticidad
Conexión Tx	Conexión para bancos y cooperativas	Alta
Conexión Tx	Campañas de conocimiento social o de ayuda de municipios.	Media alta
Conexión Tx	Campañas con fecha y hora específica	Media alta
Conexión Marketing	Campañas de marketing, con horarios regulados por el gobierno.	Media
Conexión Tx	Mensajería informativa	Media
Conexión Rx	Conexión encargada de la recepción de las respuestas del usuario. Se verifica en la operadora.	Baja

Elaborado por: Autor

3.4. Análisis UML

3.4.1. Requerimientos Funcionales.

En el siguiente apartado se describen explícitamente las acciones que debe cumplir la plataforma, y los datos que se deben contemplar para responder estas peticiones. Se plantean en base a la recopilación de datos, de apartados anteriores.

3.4.1.1. *Actores.*

Tabla 13. Actores

Actores	Jefe de TI	Encargado del manejo y administración de la plataforma.
	Personal de Soporte	Encargados del manejo y revisión de la información generada por la plataforma.

Elaborado por: Autor

Actividades del Jefe de TI

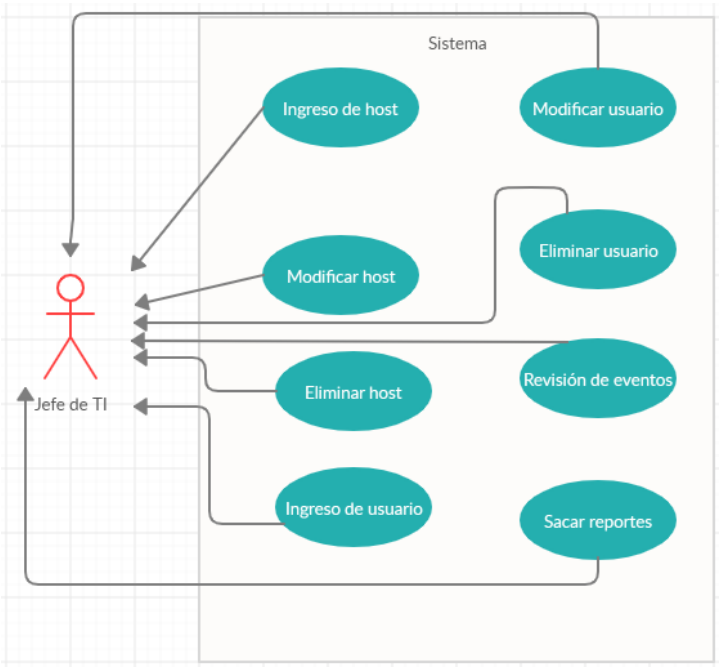


Figura 9. Actividades del Jefe de TI

Elaborado por: Autor

Tabla 14. Actividades jefe de TI

Actor	Caso de Uso	Entrada	Resultados
Jefe de TI	Ingreso de host	Ingreso campo BDD	Creación de host exitosa
	Modificar host	Modificar campo BDD	Modificación de host exitosa
	Eliminar host	Eliminar campo BDD	Eliminación de host exitosa
	Ingreso de usuario	Ingreso usuario BDD	Creación de usuario exitosa
	Modificar usuario	Modificar usuario BDD	Modificación de usuario exitosa
	Eliminar usuario	Eliminar usuario BDD	Eliminación de usuario exitosa
	Revisar eventos	Rango de fecha	Lista de eventos
	Sacar reportes	Rango de fecha	Reporte generado en web
	Generar gráficas	Identificación de host	Gráfica de host específico

Elaborado por: Autor

Actividades del personal de soporte

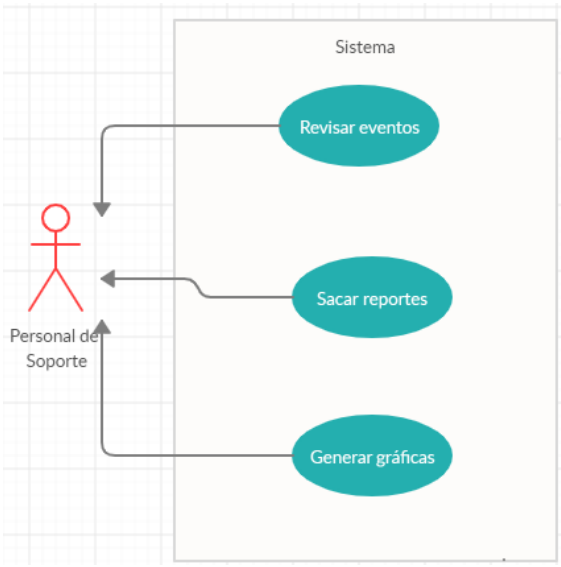


Figura 10. Actividades del personal de soporte

Elaborado por: Autor

Tabla 15. Actividades personal de soporte

Actor	Caso de uso	Entrada	Resultado
Personal de soporte	Revisar de eventos	Rango de fecha	Lista de eventos
	Sacar reportes	Rango de fecha	Reporte generado en web
	Generar gráficas	Identificación de host	Gráfica de host específico

Elaborado por: Autor

3.4.1.2. *Escenarios.*

En el siguiente apartado se describe como los usuarios actuarán con la plataforma, y como esta se comporta con esas acciones, el resultado de los mismos definirá el grado de éxito final de la implementación.

En primer lugar, se describen los escenarios de la plataforma Zabbix.

Tabla 16. Escenario 1 Zabbix, Ingreso de host

Escenario 1.	Ingreso de host
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y realiza el acoplamiento del nuevo host en el servidor de Zabbix, además de los templates necesarios para la monitorización.
Detalle.	- Autenticación en la plataforma. - Ingreso al apartado host en configuración. - Se crea el nuevo apartado del host. - Se realiza la configuración de conectividad. - Se establece el método de encriptación. - se guardan los cambios.

Elaborado por: Autor

Tabla 17. Escenario 2 Zabbix, Modificar host

Escenario 2.	Modificar host
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y realiza la modificación de parámetros de un host.
Detalle.	- Autenticación en la plataforma. - Ingreso al apartado host en configuración. - Se edita los parámetros del host requerido. - Se guardan los cambios.

Elaborado por: Autor

Tabla 18. Escenario 3 Zabbix, Eliminar host

Escenario 3.	Eliminar host
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y realiza la eliminación de un host.
Detalle.	- Autenticación en la plataforma. - Ingreso al apartado host en configuración. - Se elimina la configuración del host requerido. - Se guardan los cambios.

Elaborado por: Autor

Tabla 19. Escenario 4 Zabbix, Revisar eventos

Escenario 4.	Revisar eventos
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y revisa los eventos generados en un host específico.
Detalle.	- Autenticación en la plataforma. - Revisión de notificaciones en el Dashboard. - Selección de notificación. - Especificaciones de evento.

Elaborado por: Autor

Tabla 20. Escenario 5 Zabbix, Sacar reportes

Escenario 5.	**Sacar reportes**
Prioridad.	Media
Acción.	El Jefe de TI se identifica en la plataforma y desea generar el reporte de un grupo de host en un rango de fechas específica.
Detalle.	- Autenticación en la plataforma. - Selección del reporte requerido en la sección reportes. - Selección del rango de fechas que se desea. - Selección del grupo de host requerido. - Generación de reporte web.

Elaborado por: Autor

Tabla 21. Escenario 6 Zabbix, Generar gráficas

Escenario 6.	**Generar gráficas**
Prioridad.	Media
Acción.	El Jefe de TI se identifica en la plataforma y desea generar gráficas estadísticas de un parámetro del host.
Detalle.	- Autenticación en la plataforma. - Selección de gráficas en la pestaña monitoreo. - Selección de rango de fechas. - Selección de parámetro requerido. - Generar gráfica.

Elaborado por: Autor

Ahora se describen los escenarios posibles dentro de la plataforma Monit, encargada de la monitorización y control de procesos.

Tabla 22. Escenario 1 Monit, Ingreso de servicio

Escenario 1.	**Ingreso de servicio**
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y desea agregar un

	nuevo servicio para ser monitoreado.
Detalle.	- Autenticación en el host donde se aloja el servicio por consola. - Ingreso al path de los servicios monitoreados. - Creación de archivo de monitoreo. - Creación de archivo de control. - Reinicio del servicio.

Elaborado por: Autor

Tabla 23. Escenario 2 Monit, Modificar el servicio

Escenario 2.	**Modificar el servicio**
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y desea modificar los parámetros de un servicio.
Detalle.	- Autenticación en el host donde se aloja el servicio por consola. - Ingreso al path de los servicios monitoreados. - Modificación del archivo de configuración. - Reinicio del servicio.

Elaborado por: Autor

Tabla 24. Escenario 3 Monit, Eliminar el servicio

Escenario 3.	**Eliminar el servicio**
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y desea eliminar el servicio.
Detalle.	- Autenticación en el host donde se aloja el servicio por consola. - Ingreso al path de los servicios monitoreados. - Eliminación del archivo de configuración. - Reinicio del servicio.

Elaborado por: Autor

Tabla 25. Escenario 4 Monit, Verificación del estado de los servicios.

Escenario 4.	Verificación del estado de los servicios.
Prioridad.	Alta
Acción.	El Jefe de TI se identifica en la plataforma y desea eliminar el servicio.
Detalle.	- Autenticación en la plataforma web. - Selección de Monit en la pestaña de monitoreo. - Identificación de parámetros del servicio requerido.

Elaborado por: Autor

3.4.1.3. *Casos de uso*.

Los casos de uso se establecen en base a los escenarios planteados, y son acciones que los actores toman en las situaciones descritas con el entorno desarrollado. Ay que señalar que el sistema de monitoreo solo alerta inconvenientes de la arquitectura y que las acciones específicas en algunos problemas deben ser evaluados y solucionados por el departamento de soporte de la empresa, ya que los procesos son propios de la organización.

Tabla 26. Caso de uso 1, Reconocimiento de evento en un host

Caso de uso. 1	Reconocimiento de evento en un host
Detalle.	Al jefe de TI o personal de soporte se le dispara una alerta en su equipo de monitoreo, sobre un problema en algún host. Este se dirige a la plataforma Zabbix, se autentifica y revisa el estado del host.

Elaborado por: Autor

Tabla 27. Caso de uso 2, Desconexión de un host

Caso de uso. 2	Desconexión de un host
Detalle.	Al jefe de TI o personal de soporte se le dispara una alerta en su equipo de monitoreo, sobre la desconexión de un host. Este accede a la plataforma Zabbix, y revisa al host que se desconectó, y el tiempo de inactividad para tomar acciones.

Elaborado por: Autor

Tabla 28. Caso de uso 3, Uso elevado de la CPU en un host

Caso de uso. 3	Uso elevado de la CPU en un host
Detalle.	Al jefe de TI o personal de soporte se le dispara una alerta en su equipo de monitoreo, sobre el uso elevado de la CPU de un host. Este accede a la plataforma Zabbix y revisa el porcentaje de uso para tomar medidas.

Elaborado por: Autor

Tabla 29. Caso de uso 4, Tiempo de retraso de un host

Caso de uso. 4	Tiempo de retraso de un host
Detalle.	Al jefe de TI o personal de soporte se le dispara una alerta en su equipo de monitoreo, sobre el retardo del tiempo de respuesta de un host. Este accede a Zabbix y verifica el tiempo de retardo del host, para tomar medidas.

Elaborado por: Autor

Tabla 30. Caso de uso 5, Inactividad de un proceso

Caso de uso. 5	Inactividad de un proceso
Detalle.	Al jefe de TI o personal de soporte se le dispara una alerta en su

	equipo de monitoreo, sobre la caída de un proceso, y las acciones que se tomaron para su reinicio, este accede a la plataforma Zabbix, pestaña Monit, para verificar el estado del proceso.

Elaborado por: Autor

3.4.2. Requerimientos no funcionales.

Ahora se detalla los elementos básicos de evaluación de la plataforma, que no se ajustan a un esquema secuencial, sino al desempeño y uso que el usuario final le dé al sistema.

Tabla 31. Interfaz gráfica

N. 1	Interfaz gráfica
Detalle.	Es el elemento de interacción elemental de un sistema, y su diseño debe estar orientado a simplificar el trabajo de un equipo, además de ser configurado con la mayor simplicidad posible.

Elaborado por: Autor

Tabla 32. Protocolos de red

N. 2	Protocolos de red
Detalle.	El sistema debe soportar protocolos de red que permitan un grado se seguridad elevado, con la encriptación de contenido y detección de host sin retardos elevados.

Elaborado por: Autor

Tabla 33. Protocolos de red

N. 3	Protocolos de red
Detalle.	El sistema debe soportar protocolos de red que permitan un grado

	se seguridad elevado, con la encriptación de conexión y detección de host sin retardos elevados.

Elaborado por: Autor

Tabla 34. Base de datos

N. 4	Base de datos
Detalle.	El sistema debe alojarse en el mismo host que la base de datos, para mejorar tiempo de respuesta y uso de snap shots.

Elaborado por: Autor

Tabla 35. Sistema operativo

N. 5	Sistema operativo
Detalle.	El sistema operativo debe gestionar la plataforma y base de datos con facilidad, además de permitir la expansión y conexión con otros hosts en la misma red.

Elaborado por: Autor

Tabla 36. Reacción ante incidencias

N. 6	Reacción ante incidencias
Detalle.	La plataforma debe responder con eficiencia ante cualquier novedad en los host o en los procesos, sin dejar de notificar un problema, para evitar retardos y desestimación del entorno.

Elaborado por: Autor

Tabla 37. Número de usuarios concurrentes

N. 7	Número de usuarios concurrentes
Detalle.	La plataforma es capaz de soportar el acceso concurrente de un número de 20 sesiones, lo que satisface las necesidades de los

	involucrados con solvencia.

Elaborado por: Autor

3.5.Administración de roles

Se define el conjunto de usuarios que interactúan en la plataforma, y las actividades que realizan en el entorno. Estos son asignados por el rango y actividades desempeñadas en la organización.

3.5.1.Roles de Zabbix.

Root. Este rol es el encargado de la gestión completa del sistema Zabbix, y es el que accede al host en donde se aloja toda la plataforma. El acceso está restringido por medio de la consola, y está encargado de los ajustes elementales del sistema, así como de la gestión de la base de datos. Tiene el poder absoluto sobre toda la plataforma y puede realizar cualquier cambio, así como modificar en sí el código del sistema.

Administrador. Este rol tiene los mayores privilegios dentro del entorno gráfico de la plataforma, encargo de agregar, modificar o eliminar host, o el conjunto de ítems que se monitorean en el sistema. Además, puede gestionar la creación de usuarios para el uso de la plataforma por parte del personal de la organización.

Usuarios de soporte. Este rol es creado por el administrador en la interfaz web, y es capaz de acceder a los servicios que el administrador considere. En general no le

es posible modificar parámetros de la configuración, solo la visualización de la información.

3.5.2. Roles de Monit.

Root. Este rol es el encargado de la gestión completa del sistema Monit, y es el que accede al host en donde se aloja toda la plataforma. El acceso está restringido por medio de la consola, y está encargado de los ajustes elementales de los procesos monitoreados, así como la creación de nuevos los script para la gestión de los mismos.

Usuarios de soporte. Es capaz de acceder a la visualización del estado de los procesos por la interfaz web, pero no determina ninguna acción con respecto a estos.

Capítulo IV

4.1. Análisis detallado del entorno Zabbix

El siguiente apartado consta de la descripción de los módulos principales que forman la plataforma Zabbix, las entradas y salidas que requiere cada sección principal, y como se presenta la información a los usuarios finales.

4.1.1. Módulo de monitoreo.

Dashboard. Panel principal de la plataforma Zabbix, en donde se presenta la información más relevante del monitoreo. Presenta las especificaciones del sistema, como el servidor donde se aloja la plataforma y el puerto que requiere, además muestra el número de host que está soportando y las especificaciones para su funcionamiento. Organiza las estadísticas en gráficas organizadas por colores, en donde especifica el grado de importancia de cada color, como se muestra en la figura 11. Además, en cuadros describe las incidencias de que cada host presentó en todo el tiempo que el sistema ha estado en funcionamiento, organizadas de manera cronológica.

Posee una sección con la hora de la región que se establece en la configuración e instalación de Zabbix (puede revisar en anexos), y secciones en donde se puede establecer como favoritas un grupo de gráficas de host principales, o mapas de la topología organizacional, como se muestra en la figura 12.

Dashboard de Zabbix

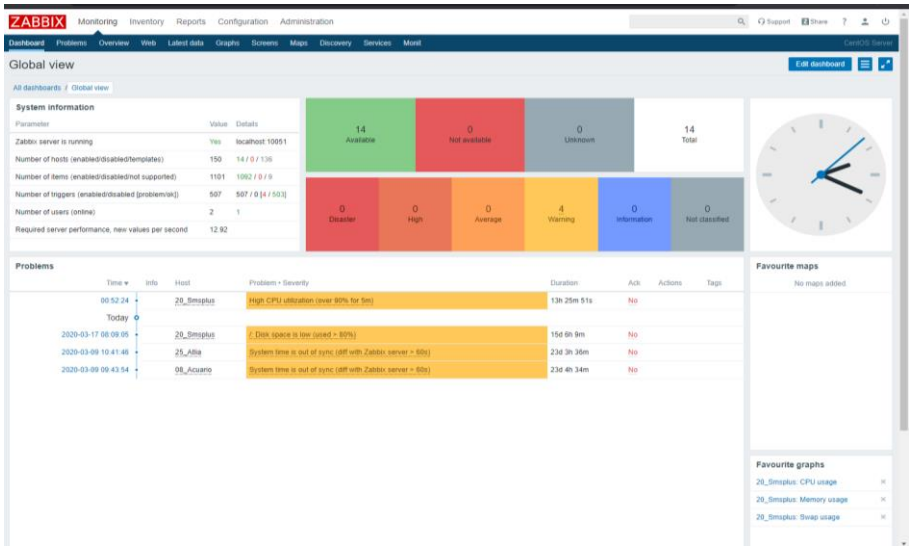


Figura 11. Dashboard de Zabbix

Elaborado por: Autor

Dashboard de Zabbix con detalles de problemas

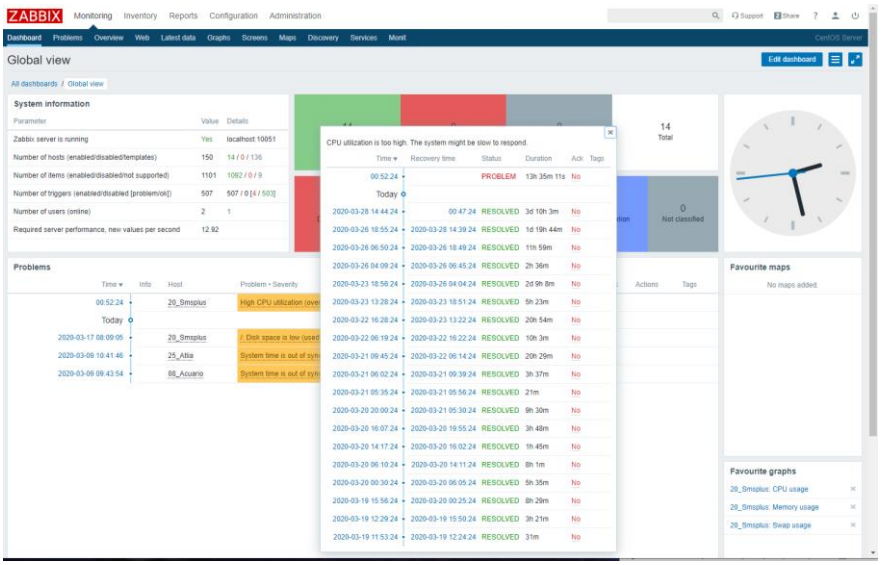


Figura 12. Dashboard de Zabbix con detalles de problemas

Elaborado por: Autor

Problems. Este módulo detalla los incidentes generados en cada host, para ser organizados en una serie de filtros que ayuden a entender la data de mejor manera. Estos incidentes pueden ser descargados en formato .csv en la misma ventana, indicado en la figura 13.

Problems de Zabbix

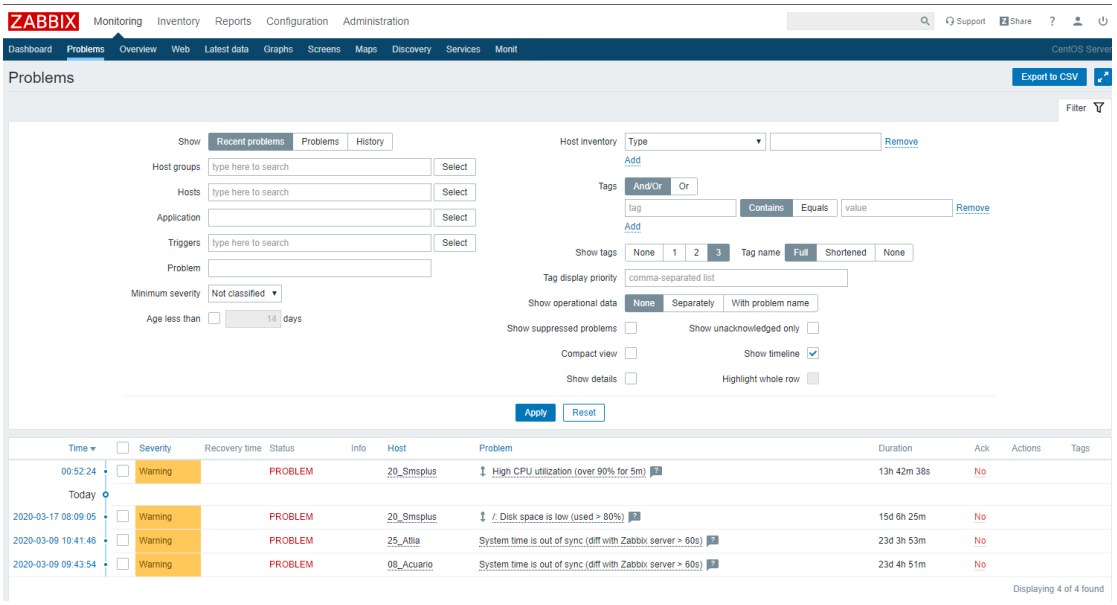


Figura 13. Problems de Zabbix

Elaborado por: Autor

Graphs. En este módulo se pueden obtener gráficas de cada host, o grupo de host. Estos pueden ser organizados en diferentes filtros y especificaciones de gráficos. Además, se maneja por períodos de tiempo muy flexibles, y rangos de colores intuitivos para el manejo de diversas gráficas a la vez, como se muestra en la figura 14.

Graphs de Zabbix

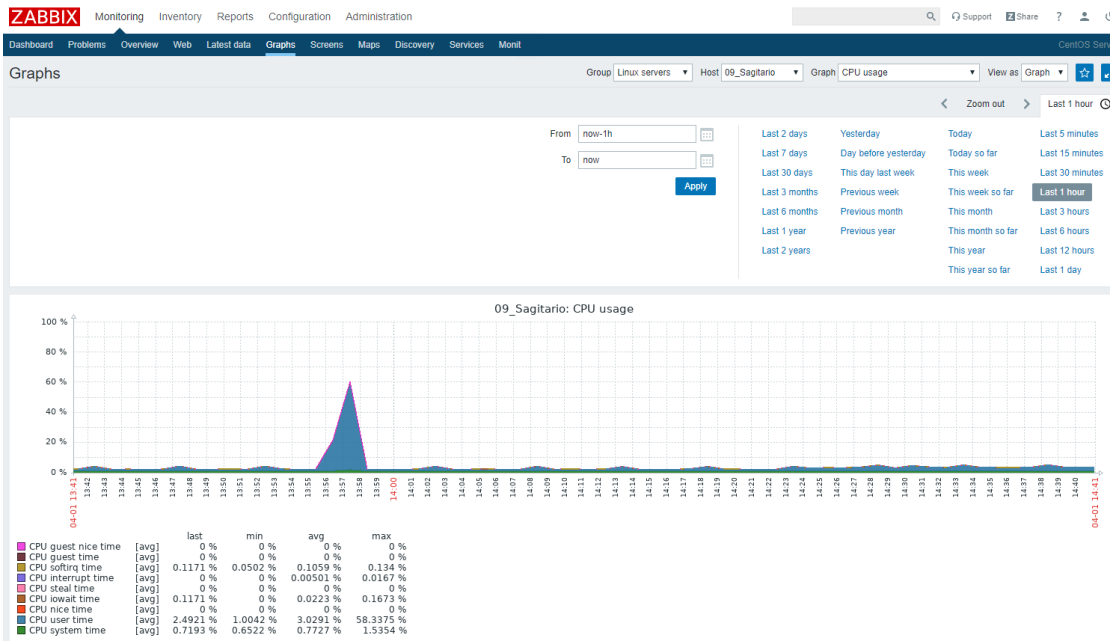


Figura 14. Graphs de Zabbix

Elaborado por: Autor

4.1.2. Módulo de inventario.

Hosts. Este módulo es el inventario de host de la plataforma, detalla los aspectos básicos de todas las terminales agregadas por el root, así como el grupo al que pertenecen, el nombre del sistema, y la versión de kernel que usan, en este caso solo se monitorean distribuciones Linux, por lo que los requerimientos son similares, indicado en la figura 15.

Hosts de Zabbix

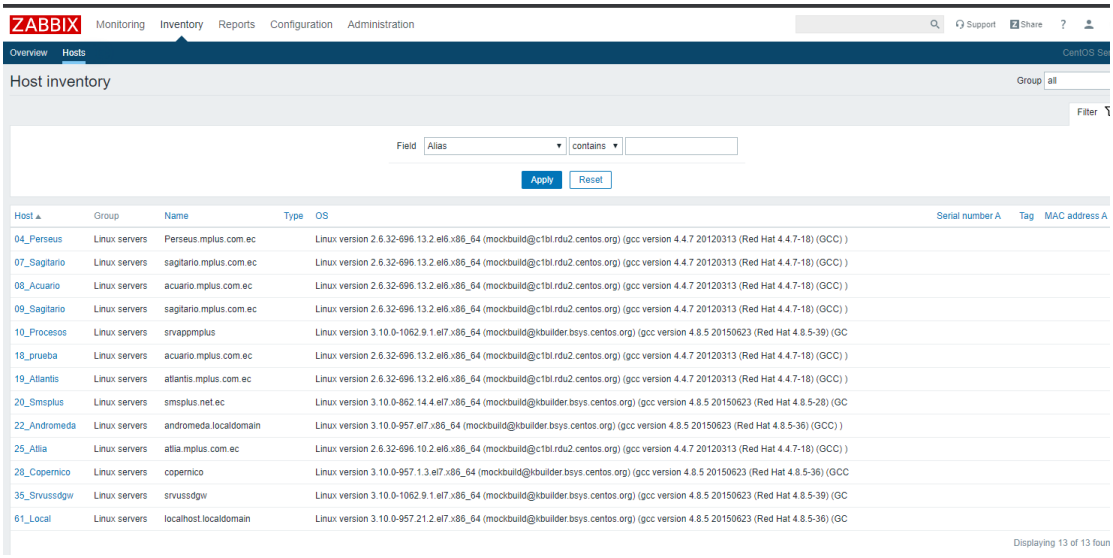


Figura 15. Hosts de Zabbix

Elaborado por: Autor

4.1.3. Módulo de reportes.

Availability report. El módulo de reportes de disponibilidad, detalla cómo las incidencias en un host, perjudicaron el tiempo de servicio al usuario, como se muestra en la figura 16. Parámetros como el uso excesivo de la memoria, el tiempo de respuesta con alto retraso o problemas que causen un desfase en la comunicación en tiempo real, entre los hosts y la plataforma. En este caso se manejan filtros de tiempo ya que cada host es agrupado en períodos de incidencia por separado

Availability report de Zabbix

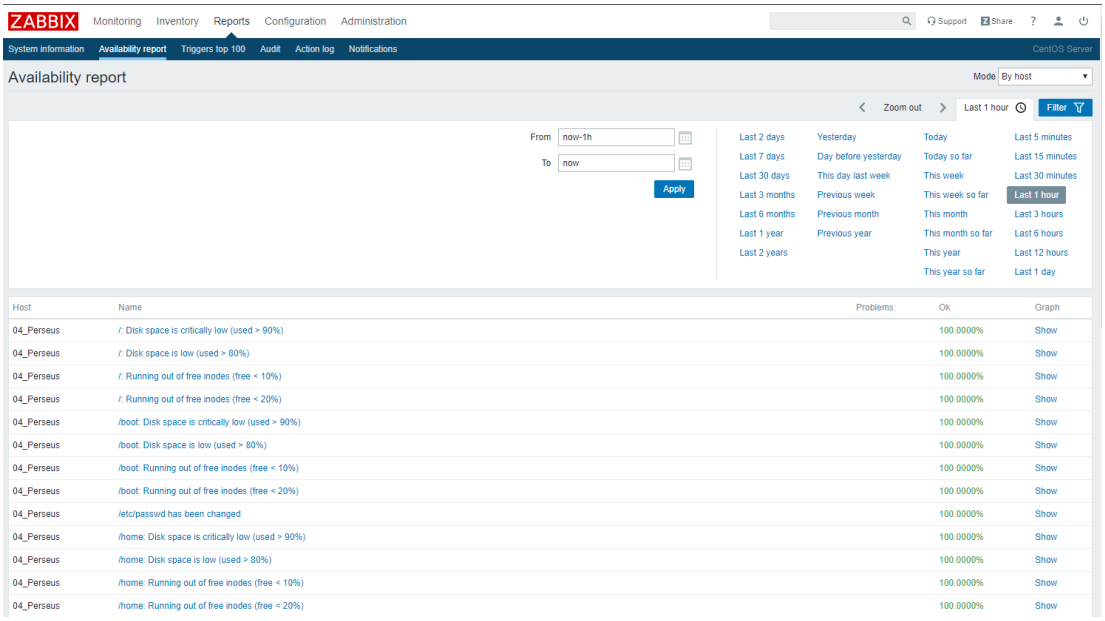


Figura 16. Availability report de Zabbix

Elaborado por: Autor

4.1.4. Módulo de configuración.

Host groups. En este módulo se accede a la configuración de los grupos, se añade o elimina configuraciones en base al sistema operativo, finalidad, entre otros parámetros que el administrador crea necesario para crear cada grupo, detallado en la figura 17. Se puede acceder a templates, que son los encargados de decirle al sistema que debe verificar en cada grupo, o que datos debe guardar en la base de datos.

Host groups de Zabbix

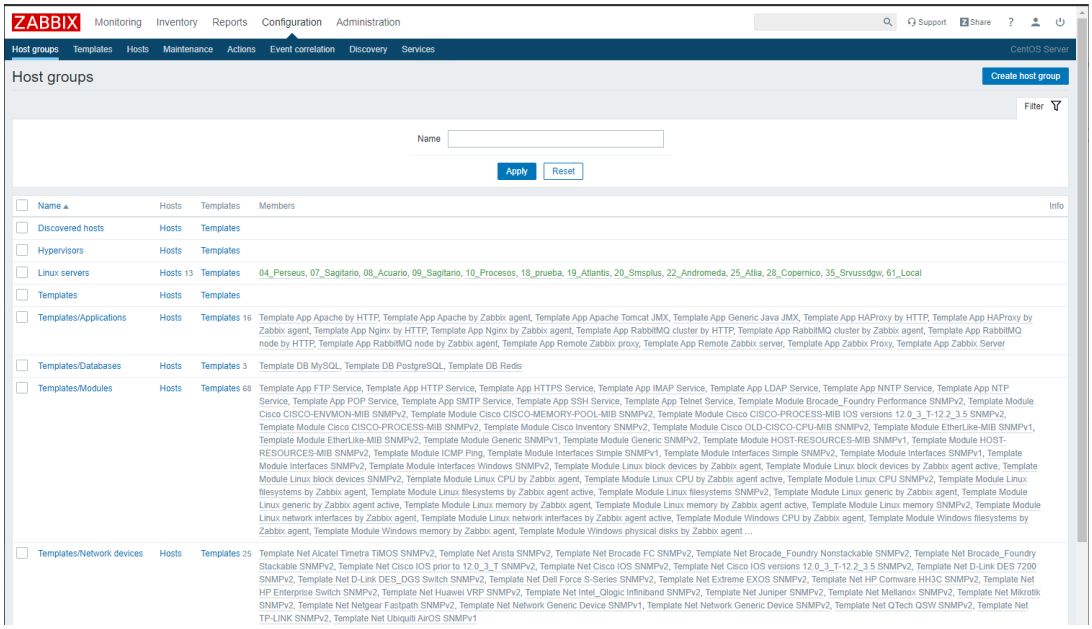


Figura 17. Host groups de Zabbix

Elaborado por: Autor

Templates. En este módulo se organizan las acciones que se van a monitorear en los servidores que usen templates específicos, se puede ver estados del sistema, la memoria, el uso de la CPU, como se muestra en la figura 18. Además, el estado de servicios entre otros parámetros que se ajustan al sistema que se usa, y que además se pueden modificar para adaptarse de mejor manera.

Hosts. Este módulo está encargado de la gestión de cada host, adicionando los clientes al servidor, editando parámetros o eliminado instancias, detallado en la figura N0. 19. Se detalla las IP de cada uno, así como el estado del cliente Zabbix, y el tipo de encriptación usada. Además, de los templates que maneja la monitorización.

Templates de Zabbix

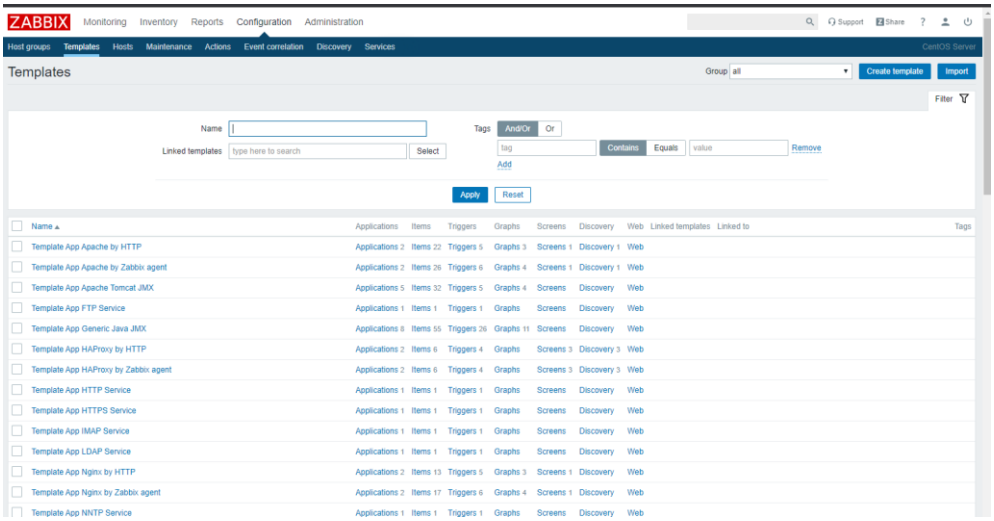


Figura 18. Templates de Zabbix

Elaborado por: Autor

Hosts de Zabbix

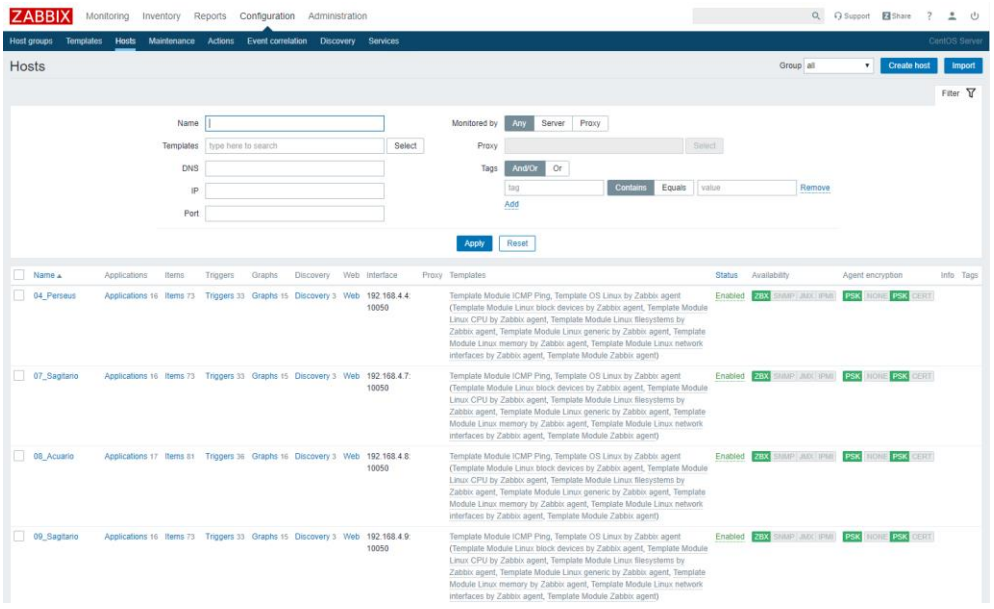


Figura 19. Hosts de Zabbix

Elaborado por: Autor

4.1.5. Módulo de administración.

En este se realizan todas las configuraciones del sistema, comenzando con las configuraciones gráficas como los temas, el tamaño de la fuente y el número de resultados que se desean ver por pantalla cargada. Sigue el apartado de configuración de proxies. La autenticación también se puede personalizar; los ajustes de HTTP con dominios propios y los ajustes de LDAP[2] con el host y puerto de acceso, y los parámetros de seguridad requeridos. Como se describe en (Zabbix, 2020)

Permite la gestión de grupos de usuarios y usuarios nuevos que pueden tener acceso al sistema, así como funciones específicas de cada uno de ellos, restringiendo funciones que no competen el tipo de administración.

Los media types, son los ajustes de los servicios implementados en la plataforma, se gestiona las notificaciones habilitadas y los script. En el siguiente apartado de Script, se puede manipular el código que conforma estos, y el Queue detalla las consultas que se definen para la base de datos fuera de las propias del sistema.

4.1.6. Notificaciones en Zabbix.

Para la gestión de las notificaciones de Zabbix se usa pushover, luego de la instalación que se puede verificar en la sección de anexos, se procede a darle los permisos de notificaciones a los usuarios pertinentes. Para esto de usa en el módulo de administración el apartado Users, en donde en el o los usuarios requeridos se agrega las credenciales pertenecientes a la API de pushover y a la API de la

[2] Protocolo de aplicación para el acceso y el mantenimiento distribuido de IT. (Bayas, 2015)

aplicación. Estas configuraciones se deben hacer en Media dentro del usuario elegido. La gestión de los pushover se hace con el envío de esas notificaciones en el móvil que se configure, esto se puede verificar en la instalación de la sección de anexos, el detalle de la conexión se verifica en la figura20.

Pushover en Zabbix

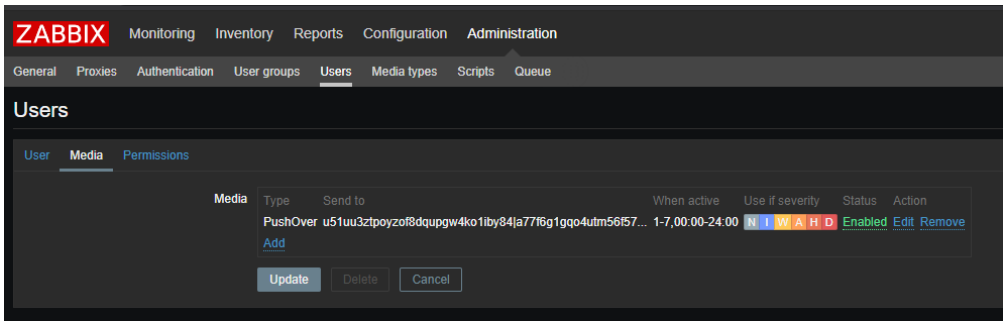


Figura 20. Pushover en Zabbix

Elaborado por: Autor

La notificación generada se gestiona en el móvil, descargando la aplicación PushOver en la Google Play o App Store, e iniciando sesión con una cuenta perteneciente a la configuración de gestión de API's. El resultado es el mostrado en la figura 21.

Notificación en PushOver

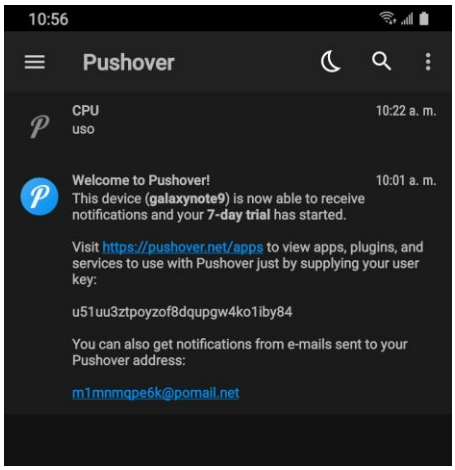


Figura 21. Notificación en PushOver

Elaborado por: Autor

4.2. Análisis detallado del entorno Monit

4.2.1. Plataforma por consola.

La plataforma Monit se instala en el servidor en donde se alojan los servicios que se requiere monitorear, luego se levanta la interfaz de la aplicación en el servidor apache de ese mismo servidor. Aun así, los servicios de Monit solo se pueden configurar por consola, la mostrada en la figura 22 y con el usuario root, debido a que la verificación de un proceso requiere el establecimiento de un script específico que controle su funcionamiento, el desarrollo de estos se detalla en la sección de anexos.

Consola Monit

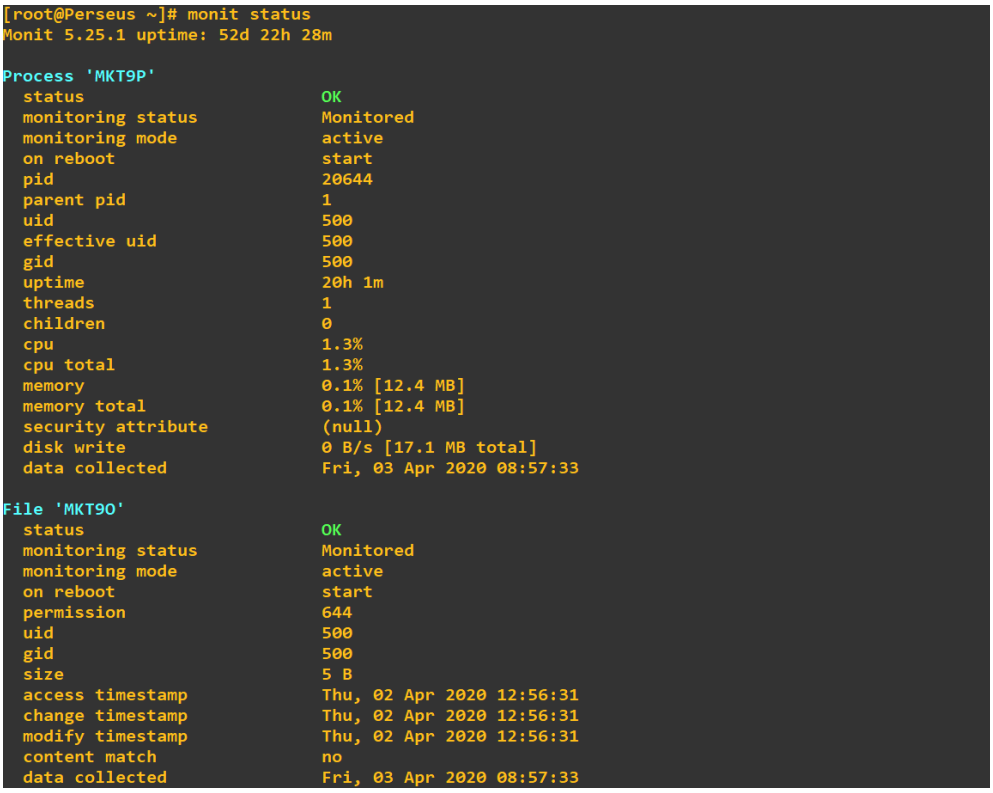


Figura 22. Consola Monit

Elaborado por: Autor

4.2.2. Interfaz Web.

La interfaz web de Monit es simple y permite la verificación visual de incidentes, con un estado operativo descrito en la segunda columna, y detallado en la figura 23. También permite ver estadísticas de tiempo de actividad desde su último reinicio, uso de CPU, la memoria que usa en el sistema y abstracciones de la memoria Swap, recalcando que toda la plataforma se maneja con host en terminales Linux.

Interfaz Monit

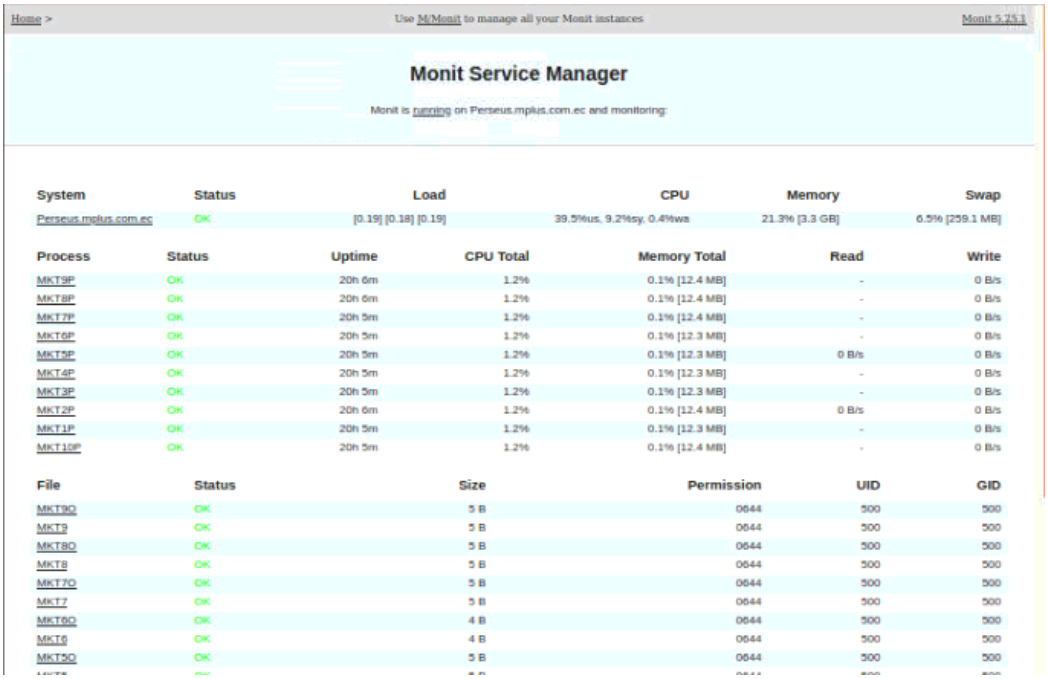


Figura 23. Interfaz Monit

Elaborado por: Autor

4.2.3. Inserción de Monit en Zabbix.

Para la implementación de la interfaz web en el entorno Zabbix, se plantea la solución de re direccionamiento, como implica los siguientes pasos.

- Acceder al código fuente de Zabbix, en este caso alojado en cd /usr/share/Zabbix.

- Ingresa al menú principal, donde se detallan las pestañas de la interfaz gráfica.
- Se identifica el módulo de monitoreo en las pestañas.
- Se crea un nuevo ítem en este módulo con el nombre Monit, figura 24.
- Para la adición de Monit se crea un url que dirige al servidor Monit.

Código Zabbix - Monit

```
$zbx_menu = [
    'view' => [
        'label' => _('Monitoring'),
        'user_type' => USER_TYPE_ZABBIX_USER,
        'default_page_id' => 0,
        'pages' => [
            [
                'url' => 'zabbix.php',
                'action' => 'dashboard.view',
                'active_if' => ['dashboard.list', 'dashboard.view'],
                'label' => _('Dashboard'),
            ],
            [
                'url' => 'zabbix.php',
                'action' => 'problem.view',
                'active_if' => ['problem.view', 'acknowledge.edit'],
                'label' => _('Problems'),
                'sub_pages' => ['tr_events.php']
            ],
            ...
            [
                'url' => 'http://192.168.4.4:2812',
                'label' => _('Monit')
            ]
        ]
    ],
```

Figura 24. Código Zabbix – Monit

Elaborado por: Autor

Finalmente, el resultado se puede verificar en la pestaña Monit, figura 25, dentro del conjunto de opciones de Zabbix.

Acceso Zabbix – Monit

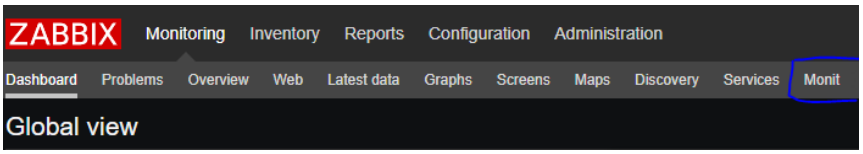


Figura 25. Acceso Zabbix – Monit

Elaborado por: Autor

4.2.4. Notificaciones en Monit.

El notificar un incidente en los procesos monitoreados por Monit, solo es una representación de ese problema, debido a esto se define que esto se realice por medio de correo electrónico. La configuración del correo se establece en la sección de anexos, pero se puede representar en un esquema de macros cuyo resultado se presenta en la figura 26.

Alerta Monit – Correo

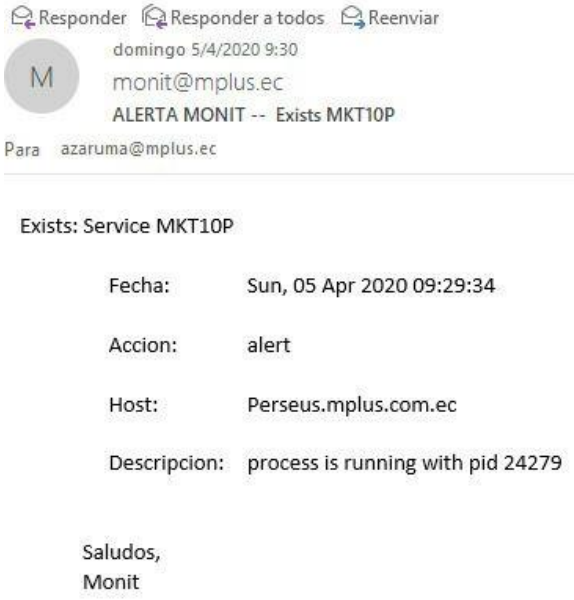


Figura 26. Alerta Monit – Correo

Elaborado por: Autor

Par las notificaciones de una importancia media – alta y alta, se define el envío de SMS, cuyo script se detalla en la sección de anexos, pero cuyo resultado es el envío de una plantilla defina en la empresa, figura 27 y aprobada por las operadoras del país.

Alerta Monit - SMS

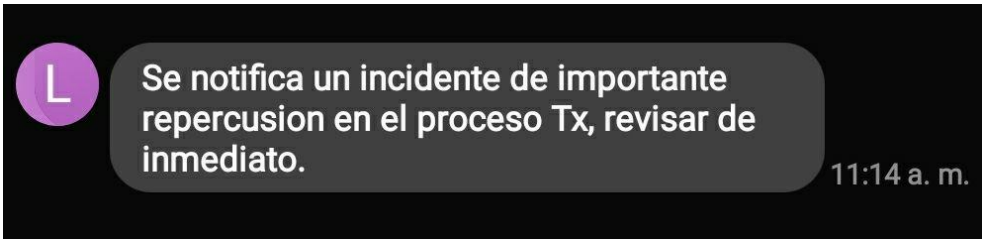


Figura 27. Alerta Monit – SMS

Elaborado por: Autor

Para la realización de las pruebas se generará ambientes controlados que simulen situaciones de estrés para las terminales, que reflejen en acciones y notificaciones de la plataforma. Para este apartado es importante saber, que el sistema puede tomar comportamientos distintos en su uso en producción, y las simulaciones son de carácter educativo, así que no se debe usar estos apartados para réplicas en otros sistemas.

4.3. Aplicación para entorno Android

La aplicación para teléfonos inteligentes basados en Android, se implementó en el IDE Android Studio, (el código fuente se puede revisar en la sección de anexos). La estructura de la app consta de un visor web que genera vistas de la página de Zabbix, lo que permite cargar estas en la cache del dispositivo y mantenerlas abiertas de manera independiente, lo que no se podría con el uso de un navegador móvil. Además, se usa un cliente que maneja los certificados de la web, para evitar el

manejo de mensajes del visor y cargar la página en menor tiempo, interfaz de entrada en la figura 28.

Aplicación en Android

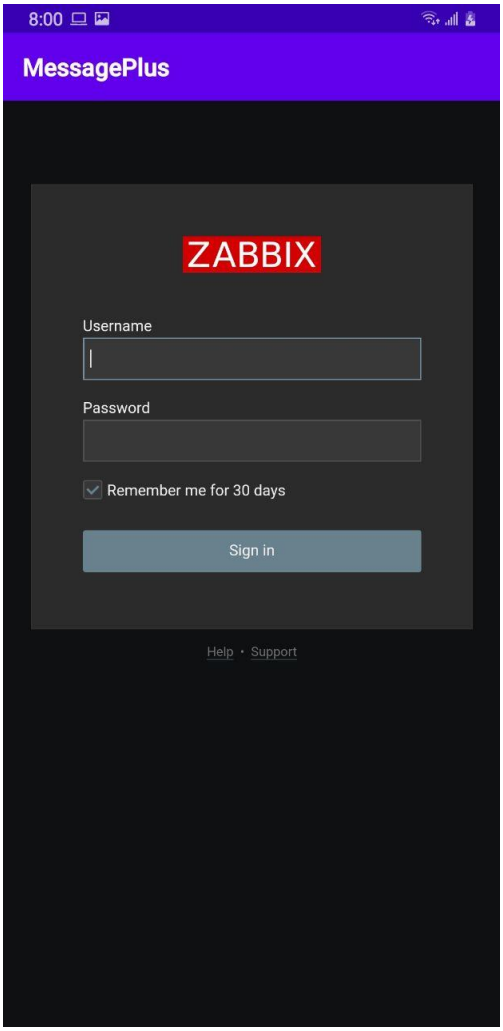


Figura 28. Aplicación en Android

Elaborado por: Autor

Capítulo V

5.1. Pruebas y Resultados

Una vez implementada la plataforma se procede con las pruebas de las principales métricas de Zabbix y Monit. Además, de la especificación de las actividades e información que se visualiza en la pantalla. Muchas de estas pruebas se realizaron en situaciones reales puesto que las alertas se desarrollan mientras la empresa está ofreciendo servicios. Las pruebas más críticas se desarrollan en un ambiente controlado sin interferir en las actividades de la organización, todo esto se detalla en cada apartado. Ahora se detallan los resultados.

5.1.1. Prueba de espacio del disco.

La siguiente prueba muestra el estado de un servidor de base de datos, al que se le han acumulado cerca de 4 millones de registros de envíos de SMS e IVR en un mes, por lo que ahora se le va a realizar una copia de seguridad y la base se limpia para nuevos registros del nuevo mes, el estado del espacio del disco se muestra en la figura 29.

Resultado de espacio en disco

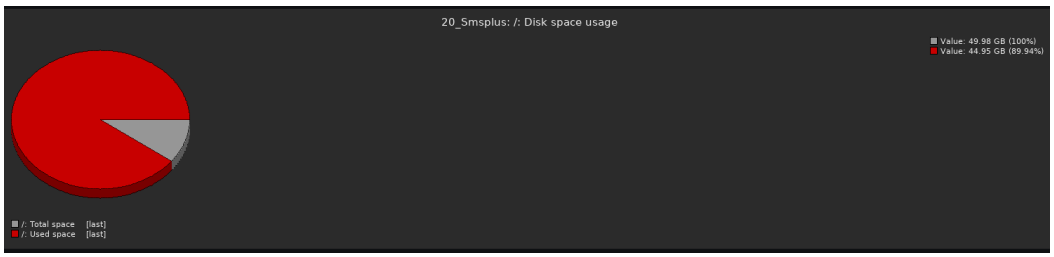


Figura 29. Resultado de espacio en disco

Elaborado por: Autor

Pese a que el uso del disco duro llegó a un 89.95 % la empresa contempla esta gran cantidad de datos y establece un estado de alarma si los niveles llegan a sobrepasar un 95 %, en donde se puede ver comprometida la base de datos, es por esto que el sistema o notifica como un estado de alerta, solo como una advertencia del uso del disco duro.

5.1.2. Prueba de saltos de la CPU.

Los sistemas operativos están sujetos a verificar una gran cantidad de procesos que se presentan de manera simultánea, por ello el control de los llamados saltos o llamadas a la central de procesos es indispensable. La siguiente prueba se realiza en 08_Acuario, un servidor dedicado a responder solo ciertos tipos de campañas o recargas telefónicas, por lo que su CPU solo es usada en momentos específicos, pero en horas picos es altamente requerida en procesamientos específicos. Este caso es de un lunes en hora pico de (9 a 13 horas) y el uso se representa en la figura 30.

Resultado de saltos de la CPU

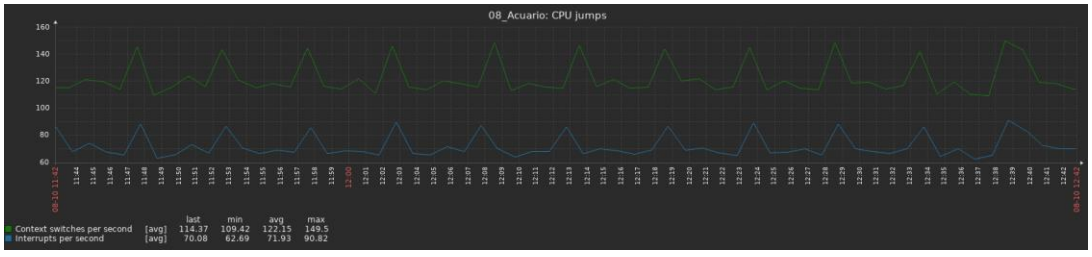


Figura 30. Resultado de saltos de la CPU

Elaborado por: Autor

Como se muestra en la gráfica anterior, los saltos representan un uso repentino de la CPU cuando esta se encuentra en reposo, en el caso de una hora pico, los cambios son más seguidos y de una duración no tan amplia por el tipo de servicio que maneja el servidor. Estos saltos no muestran una alerta pues si existe un número excesivo de saltos, saltaría la alerta de la carga de la CPU.

5.1.3. Prueba de carga de la CPU.

La siguiente prueba se realiza en la carga de un conjunto de datos a la base de datos, que corresponde a un envío masivo de SMS en una hora pico (de 8 a 12 del día), se constituye de la carga de una campaña de 10 000 SMS, que son despachados hacia las operadoras desde el servidor 04_Perseus, las métricas del uso de la CPU se muestran en la figura 31.

Resultado de uso de la CPU

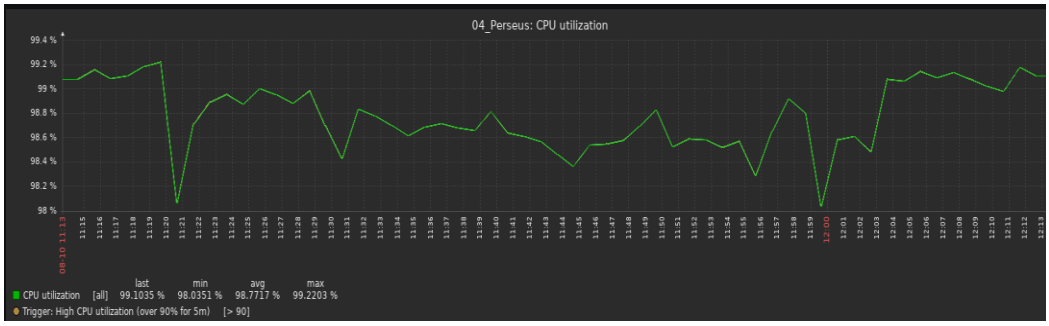


Figura 31 a. Resultado de uso de la CPU

Elaborado por: Autor

Como se muestra el uso de la CPU sobrepasa el 90 % en algunos picos importantes, lo que desencadena la notificación del sistema, que mantiene el estado de alerta en

todo el intervalo de tiempo donde los picos suben y bajan del nivel indicado. Cuando el estado de la CPU vuelve a ser normal, el estado de alarma cambia, como se muestra en le figura

Notificación de uso de la CPU

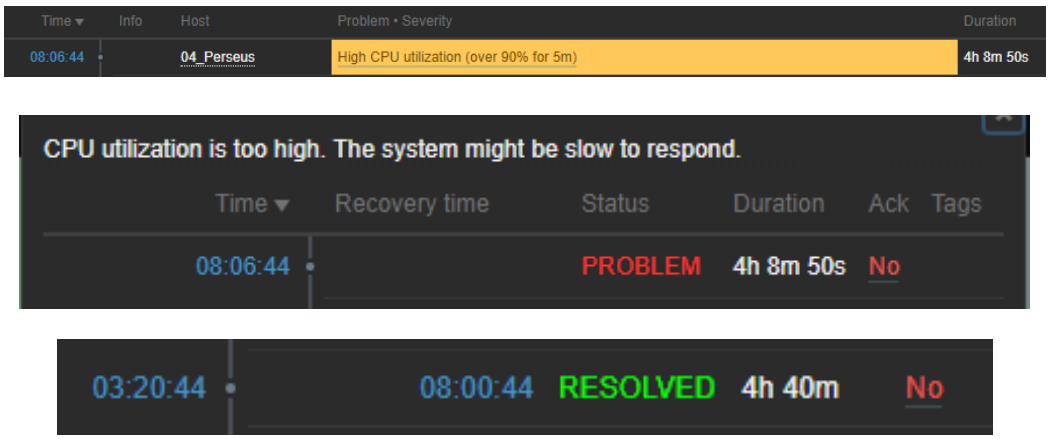


Figura 32. Notificación de uso de la CPU

Elaborado por: Autor

5.1.4. Prueba de tráfico de red.

La siguiente prueba se realiza al tener una carga grande desde la web de un cliente hacia el servidor, en este caso, la carga de una campaña de SMS, en donde se insertan por medio de un .csv un conjunto de 5 000 registros que ingresan a la base de datos, los resultados se detallan en la figura 33.

Resultado de tráfico de red

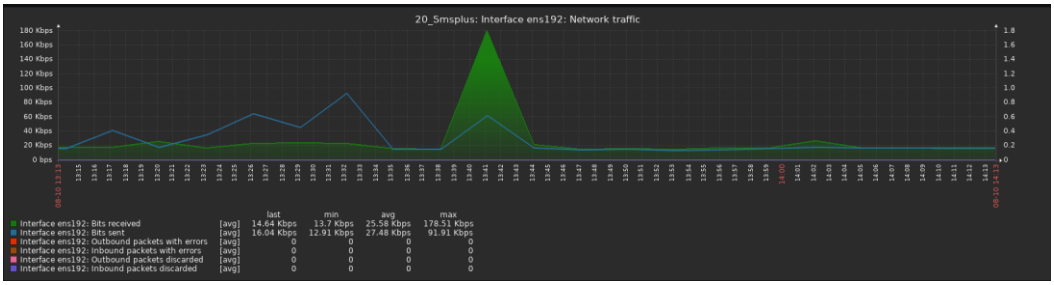


Figura 33. Resultado de tráfico de red

Elaborado por: Autor

Como se puede apreciar existe un pico de uso en el tráfico de la red de 20_Smsplus por la interfaz habilitada, esto se debe al incremento del tráfico de datos generado por la carga de la campaña en el sistema, aun así, el servidor está preparado para la gestión de estos incidentes y lo solventa de manera satisfactoria, donde como resultado que el pico no se extienda demasiado tiempo, y la red no se sature.

5.1.5. Prueba de uso de la memoria RAM.

Al manejar ambientes virtuales, la memoria RAM depende mucho de la carga del sistema, en el siguiente caso se muestra a 04_Perseus, que es una terminal encargada del manejo y despacho de SMS para la gestión de campañas masivas, por lo que su memoria RAM está en un uso aceptable, también influye el hecho de que el servidor lleva 3 meses sin ser reiniciado, debido a que el análisis muestra que el uso de memoria es constante y no presenta picos anormales que puedan influir en su funcionamiento (información de parte de soporte), el resultado se muestra en la figura 34.

Resultado del uso de la memoria RAM

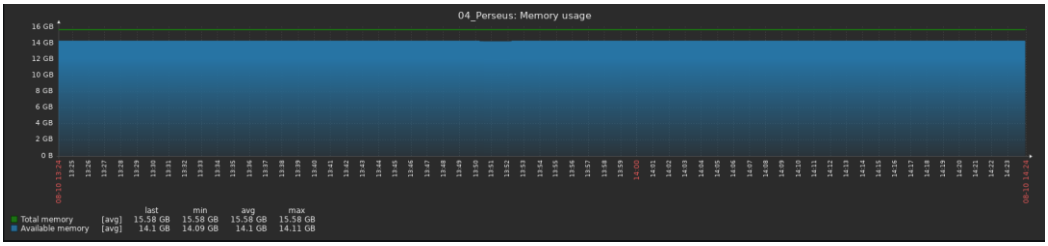


Figura 34. Resultado del uso de la memoria RAM

Elaborado por: Autor

Muestra un uso regular cerca del tope asignado al servidor, por lo que no requiere de un aumento de memoria ni de acciones de soporte.

5.1.6. Prueba de control de Procesos.

Para el control del número de procesos activos en un servidor, se presenta el caso de 08_Acuario, un servidor dedicado al manejo de recargas electrónicas por SMS, que presenta un grupo constante de procesos excepto cuando se activa una recarga en un tiempo específico, el cual hace su función y se desactiva, el resultado se muestra en la figura 35.

Resultado de control de proceso

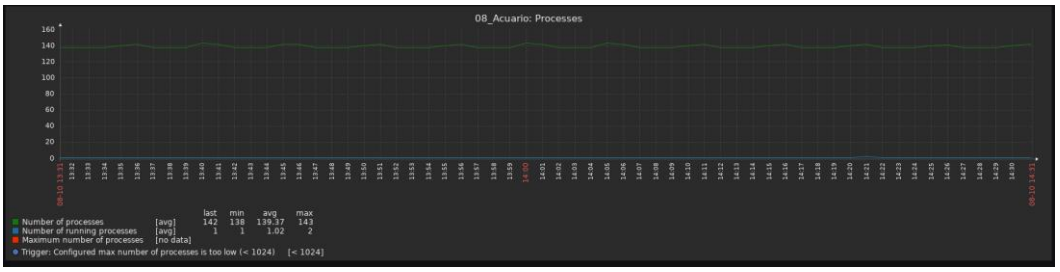


Figura 35. Resultado de control de proceso

Elaborado por: Autor

La gráfica muestra cómo se generan picos pequeños de incremento del número de procesos, debido a el análisis de la prueba del párrafo anterior. Pero este dato solo permite verificar que los procesos se estén ejecutando, aun así, no se controla por alerta debido a que este número puede subir o bajar por acciones del servidor o del personal.

5.1.7. Prueba de partición /sda.

Las /sda, son las particiones que tiene el disco con distintos propósitos, pero en este apartado se analiza los retardos de espera, el tiempo de lectura y escritura, las solicitudes de uso de consultas y el espacio de almacenamiento, para el servidor

25_Atlia, encargado del manejo de las pruebas, de nuevas funcionalidades o centros de mensajes para los clientes. Esto presenta un uso prolongado de los apartados descritos de la /sda, por lo que los resultados se muestran en las figuras No. 36, 37 y 38.

Resultado de promedio de espera del disco

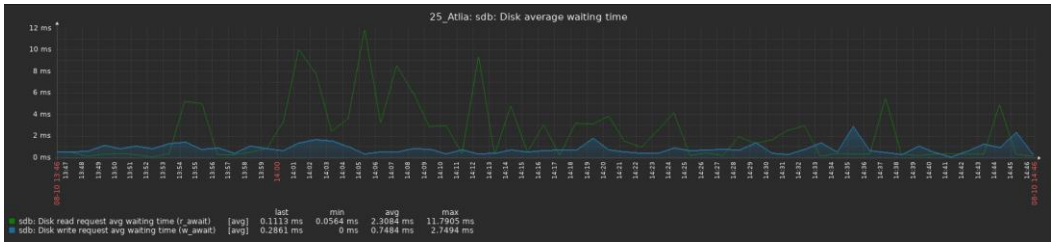


Figura 36. Resultado de promedio de espera del disco

Elaborado por: Autor

Resultado de tarifas de lectura y escritura del disco

Figura 37. Resultado de tarifas de lectura y escritura del disco

Elaborado por: Autor

Resultado de utilización y consultas al disco

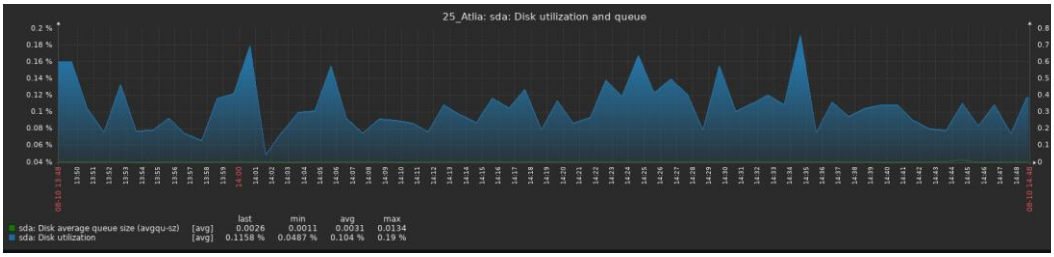


Figura 38. Resultado de utilización y consultas al disco

Elaborado por: Autor

Estas métricas verifican el estado de las particiones que Linux maneja, para que su funcionamiento sea óptimo y no se esté sobre exigiendo al sistema, lo que se evidencia por los picos que no saturan al servidor, solo denotan que se usa más en ciertas horas del día.

5.1.8. Prueba del uso de la memoria swap.

La especificación de la memoria swap, es primordial en la arquitectura de sistemas Linux, pero en términos generales es ese espacio de memoria destinado al almacenamiento de procesos que deben estar corriendo, como los demonios. Para este caso se analiza 07_Sagitario que tiene un número definido de 140 demonio funcionando en el sistema, que no deben saturarse, pero tampoco deben apagarse, pues son requeridos en cualquier momento, los resultados se muestran en la figura 39.

Resultado de uso de la swap

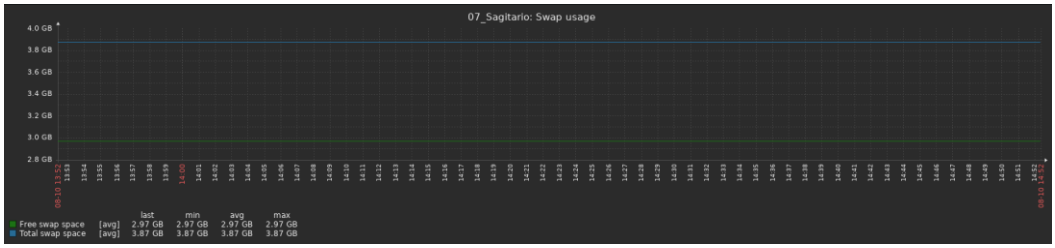


Figura 39. Resultado de uso de la swap

Elaborado por: Autor

A pesar de tener un espacio asignado de 4 GB el sistema solo requiere 3 GB para funcionar de manera correcta, y mantener sus demonios activos, por eso su gráfica es lineal sin picos de uso.

5.1.9. Prueba de perdida de conexión de una terminal.

En esta prueba se simula la perdida de conexión de una terminal de prueba, en donde por un periodo de 5 minutos se desconecta de la red principal de las demás terminales. Los resultados se presentan en la figura N. 40.

Resultado de perdida de conexión de la terminal

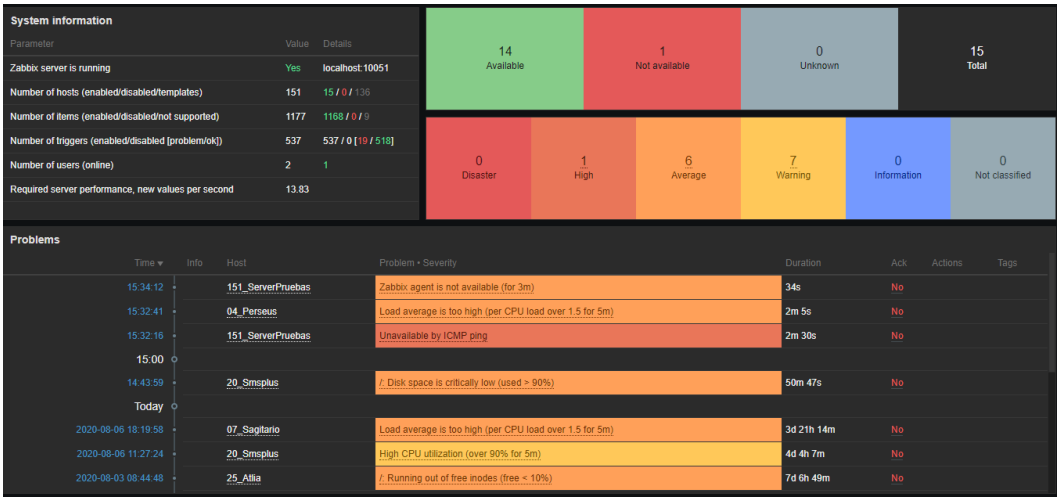


Figura 40. Resultado de perdida de conexión de la terminal

Elaborado por: Autor

Al detectar que la conexión se pierde con el servidor la plataforma genera la alerta de ICMP ping, avisando que no es posible llegar al servidor, luego cambia a una alerta de nivel alto debido a que el servidor no responde por más de 5 segundos y genera un nuevo conjunto de alertas que se verifican cada 5 minutos hasta que la conexión regrese, y el agente se comunique con el servidor. Lo que se muestra en la figura N. 41.

Resultado de reconexión de la terminal

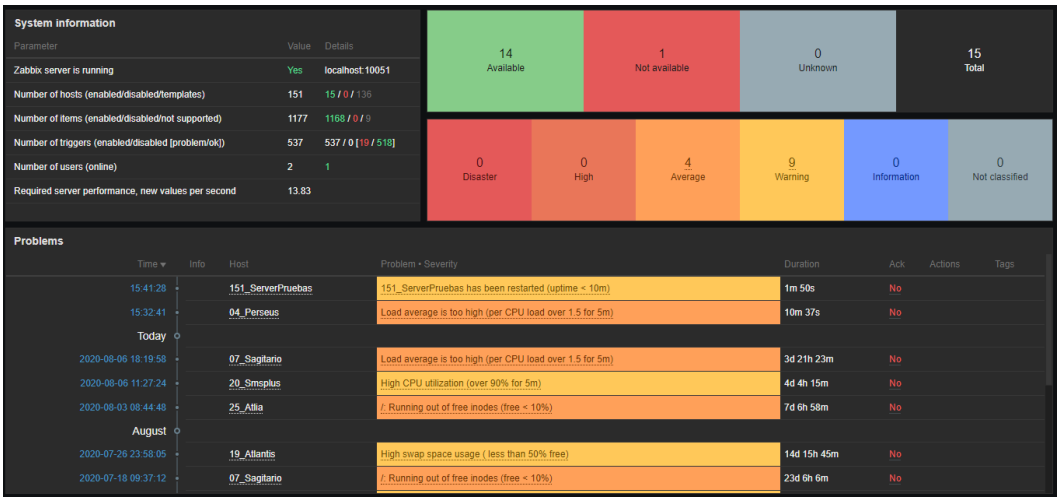


Figura 41. Resultado de reconexión de la terminal

Elaborado por: Autor

5.2. Análisis de costos

A continuación, se evalúalos costos relacionados a la implementación, monitoreo, notificación y mantenimiento de la plataforma. Posteriormente se compara con soluciones de software como servicio y software de código licenciado, para tener un contexto de los beneficios y complicaciones del código abierto. Esto en un plazo de 1 año.

5.2.1. Análisis de Zabbix y Monit.

Tabla. Análisis de costo de Zabbix y Monit

Zabbix y Monit	Costo
Costo de licencia	$ -
Costo de ejecución	$ -

Conocimiento necesario para su manejo (Curso. Zabbix de principiante a experto Udemy)	$ 40
Conocimiento para manejo de Monit (Internet)	$ 20
Costo por mantenimiento (H/H)	$ 60
Otros gastos	$ 100
Total.	$ 220

Elaborado por: Autor

5.2.2. Análisis de Acronis Monitoring Service.

Tabla. Análisis de costo de Acronis Monitoring Service

Acronis Monitoring Service		Costo
Costo Mensual	Ventajas Incluye la visualización de la infraestructura de TI en la nube. El software se mantiene en servidores externos por lo que no causa un gasto extra a la empresa en equipos. Se implementa con instaladores y su conexión entre terminales se hace en una aplicación web que se puede usar en cualquier dispositivo. Se puede implementar en muchos sistemas operativos, redes y servicios diversos.	$ 37.98

	Desventajas Uso obligatorio de internet en la infraestructura para su monitoreo. La escalabilidad tiene costos extras.	
Mantenimiento	Si se requiere de la expansión de la infraestructura sube el costo mensual.	$ -
Total (12 meses).		$ 455.76

Elaborado por: Autor

Tomado de: (Acronis, 2020)

5.2.3. Análisis de LogicMonitor.

Tabla 38. Análisis de costo de LogicMonitor

LogicMonitor		Costo
Costo anual	Ventajas Es la solución más avanzada para el monitoreo de grandes redes, basadas en SaaS. Posee un grupo de agentes basados en java que se ejecutan en la red, lo que permite una gran gama de compatibilidad. Posee un sistema de big data capaz de interpretar datos para dar predicciones sobre el comportamiento de la infraestructura. Plataforma única de acceso concurrente y escalable.	$ 4500

	Desventajas Acceso a la red en la infraestructura para el monitoreo.	
Total.		$ 4500

Elaborado por: Autor

Tomado de: (Logic Monitor, 2020)

5.2.4. Análisis de ManageEngine.

Tabla 39. Análisis de costo de ManageEngine

ManageEngine		**Costo**
Costo Mensual	Ventajas Gestión y monitorización completa, con el esquema cliente – servidor, que se instala y se usa internamente. Gran compatibilidad con marcas reconocidas de dispositivos, además que permite respaldos en la nube y una web de control externo.	$ 795
Total (12 meses).		$ 9540

Elaborado por: Autor

Tomado de: (Manage Engine, 2020)

5.2.5. Análisis de SolarWinds.

Tabla 40. Análisis de costo de SolarWinds

SolarWinds		Costo
Costo anual	Ventajas Especializado en empresa medianas, que requieran un control de su infraestructura, con vistas integrales y una plataforma de monitoreo completa. Se implementa en entornos físicos y virtuales, además de permitir el monitoreo de aplicaciones. Permite una instalación muy sencilla con la opción de descubrir los elementos de red.	$ 2995
Total.		$ 2995

Elaborado por: Autor

Tomado de: (Solar Winds, 2020)

5.2.6. Resultado de la comparación.

Una vez revisados los aspectos más relevantes de software como servicio y software licenciado, y teniendo presente los costos en que puede incurrir la plataforma implementada en este documento, es claro que el costo del servicio es más elevado en estas soluciones. Pero como se puede justificar costos como este, es claro que depende de la arquitectura de la empresa, en el análisis de costo de Zabbix y Monit no se toma en cuenta aspectos como la infraestructura y mantenimiento de esta,

debido a que la empresa posee su propio hardware destinado a otros fines, pero con la capacidad de mantener esta nueva arquitectura. Debido a esto los costos se recortan, por el contrario, una empresa que no cuente con los recursos y debe invertir en ellos para levantar una solución de monitoreo, puede recurrir a soluciones de pago que pueden representar una mejor inversión a largo plazo.

Finalmente, en el apartado de la seguridad, es importante recalcar que la plataforma en sí usa cifrado en sus conexiones, como se especifica en los anexos, pero el hecho de salir a la web por una IP pública conlleva a la verificación de los servidores de manera constante, en este caso no fue un problema debido a que la organización, tiene sus servidores en un ambiente controlado por un contratista que además les provee de un firewall y manejo de incidentes dedicado. Pero en otra empresa si es un apartado a considerar, por los riesgos de seguridad que conlleva el manejar un sistema de esta manera, gastos que pueden llegar a ser muy representativos y alteren esta comparativa de forma drástica.

CONCLUSIONES

- La infraestructura lógica de MessagePlus se desglosa en un diagrama de flujo que muestra las actividades de cada servidor y sus detalles de red en un entorno similar al que actualmente se consigue con al módulo mapas de la plataforma.
- El análisis de los servicios a ser monitoreados junto con una entrevista con el gerente de TI permitió jerarquizar y establecer el sistema de notificaciones al que se deben acoger.
- La entrevista al gerente de TI y las especificaciones de la organización. definieron los procesos que pueden ser automatizados, usando la herramienta Monit por medio de script.
- Mediante la evaluación de un grupo de herramientas de código libre, y el conocimiento de las necesidades de la organización, se escogió a Zabbix y Monit como las mejores opciones para la implementación de la plataforma web.
- La plataforma web se puso en funcionamiento en la infraestructura propia de la organización, en un entorno que une Zabbix y Monit, en donde se detallaron los servidores y los servicios principales a ser monitoreados.
- El script monitorea los logs verificando el estado del proceso y en caso de encontrar un signo de fallo de la conexión, se encarga de eliminar los residuos de esa comunicación, generando una nueva en base al código que rige los parámetros del enlace.
- La aplicación de monitoreo de servidores y servicios se implementó bajo el apartado de visor web, porque así la plataforma hace uso de un bloque

específico de cache que evita la recarga excesiva de la web, disminuyendo las peticiones al servidor y el uso innecesario del ancho de banda.

- Una vez implementada la plataforma y después de un tiempo de funcionamiento se realizaron las pruebas en el ambiente de producción, dentro de entornos controlados por la organización, como son el estado de red de los servidores, uso de disco, uso de memoria, caída de procesos entre otros apartados que se detallaron en la sección de pruebas.
- Tomando como referencia un año de operación. se evaluaron, la solución implementada frente al costo de plataformas licenciadas bajo el modelo de suscripción. Evidenciando las ventajas de la ejecución de software libre en una organización que cuenta con personal capacitado en TI, que puede usarlo y mantener el servicio sin incurrir en nuevos gastos, además de no tener que sacrificar la seguridad de la empresa por compartir datos de monitoreo con terceros.
- El uso de software libre no es solo la reducción de costo a corto plazo, se debe entender como la gestión de herramientas creadas para facilitar actividades pero que requiere de un grado de conocimiento técnico para su uso y mantenimiento.
- Al unir dos herramientas de código libre, se deben entender las bases de su funcionamiento, y evaluar la efectividad de una unión por núcleo o la adaptación de un apartado en una de ellas que dirija a los detalles de la otra.

RECOMENDACIONES

- Se recomienda realizar una copia de seguridad para salvaguarda datos y ayudar a la continuidad de la operación del negocio.
- Se recomienda para una posterior mejora a la plataforma trabajar con los templates de Zabbix, profundizando en sus aplicaciones.
- Se recomienda incluir en la política de la empresa al realizar conexiones con asociados sobre una red pública, levantar una VPN para proteger los datos.

GLOSARIO DE TÉRMINOS

Mensajería masiva. envío de SMS a grandes bases de datos de usuarios a los que se les ofrece un servicio. (Message Plus S.A., 2019)

Mailing. Es la segmentación, envío y rastreo sencillo de campañas por correo electrónico a volúmenes grandes de usuarios. (Message Plus S.A., 2019)

Recargas. Plataforma de venta de recargas de telefonía celular. (Message Plus S.A., 2019)

Servicio de IVR. Servicio de llamadas automáticas y programadas de manera personalizada a bases de usuarios. (Message Plus S.A., 2019)

Analítica de big data. es la examinación de grandes volúmenes de datos para encontrar patrones que no son evidentes. (SAS, 2019)

Rack de servidores. Grupo de servidores conectados para compartir su hardware como una unidad común. (Álvarez, 2015)

Plataforma de monitoreo. Sistema de visualización de herramientas que controlan un sistema. (Álvarez, 2015)

RRD. base de datos Round-Robin (Kumar, 2020)

LISTA DE REFERENCIAS

Alvarez, A. (2015). *Análisis, diseño e implmentación de una herramienta de monitoreo y control de Datacenter basado en herramientas open source. Aplicado al banco de Guayaquil (Tesis de Pregrado).* Universidad Politécnica Salesiana Sede Guayaquil, Ingeniería ded sistemas. Guayaquil. Universidad Politécnica Salesiana . Recuperado el 2019, de https.//dspace.ups.edu.ec/bitstream/123456789/10298/1/UPS-GT001194.pdf

Aquino Quiñonez, J. C. (2017). *Manual de Instalación, Configuración de un Sistema de Gestión y Monitoreo de Redes Informáticas para Pequeñas y Medianas Empresas en El Salvador, utilizando software libre (Tesis de Pregrado).* Universidad Tecnol+ogica del Salvador, FACULTAD DE INFORMATICA Y CIENCIAS APLICADAS.

Bayas, J. (2015). *Servidor de control de dispositivos y servicios mediante el protocolo snmp para la red de datos en celec .e.p. unidad de negocio hidroagoyan (Tesis de pregrado).* Universidad Técnica de Ambato, Facultad de ingeniería en sistemas electrónica e industrial. Ambato. Universidad Técnica de Ambato. Recuperado el 2019, de http.//repositorio.uta.edu.ec/jspui/bitstream/123456789/13063/1/Tesis_t1035ec.pdf

Carlos Velasco, G. C. (2017). *Implementación de un sistema de monitoreo de redes utilizando herramientas open source y proveer servicios de directorio a través de active directory en la facultad de filoofía, letras y ciencias de la educación de la universidad de guayaquil (Pregrado).* Univerisdad Politécnica Salesiana Sede Guayaquil, Ingeniería de Sistemas. Guayaquil.

Univerisdad Politécnica Salesiana. Recuperado el 2019, de https.//dspace.ups.edu.ec/bitstream/123456789/13474/1/UPS-GT001824.pdf

Carrera Ponce Carlos, P. S. (2017). *Implementación de una herramienta de monitoreo de hardware para los servidores de ventanillas pertenecientes al banco dedl pacífico a nivel nacional. (Tesis ded Pregrado).* Universidad Politécnica Salesiana Sede Guayaquil, Ingeniería de Sistemas. Guayaquil. Universidad Politécnica Salesiana. Recuperado el 2019, de https.//dspace.ups.edu.ec/handle/123456789/15352

CentOS Org. (09 de marzo de 2020). *CentOS Org*. Obtenido de CentOS Org. https.//www.centos.org/

Daniel, V. (2019). *Platzi*. Obtenido de Platzi. https.//platzi.com/blog/metodologia-scrum-fases/

Danny, O. Y. (2016). *Implementación de herramientas de software open source para monitorear los sistemas de comunicación de voz y datos de las empresas Tuval S.A., Dimulti S.A. y Castek S.A. (Tesis de Pregrado).* Universida Politécnica Salesiana Sede Guayaquil, Ingeniería de Sistemas. Guayaquil. Universida Politécnica Salesiana. Recuperado el 2019, de https.//dspace.ups.edu.ec/handle/123456789/12294

Edison, B. M. (2013). *Análisis e implementación de un sistema de manejo de incidentes con funcionalidad extendida notificación de correo electrónico bajo gnu/linux aplicado a los servidores y enlaces lan y wan de la empresa Edesa s.a. (Tesis de Pregrado).* Universidad Politécnica Salesiana Sede Quito, Ingeniería de Sistemas. Quito. Universidad Politécnica Salesiana . Recuperado el 2019, de https.//dspace.ups.edu.ec/handle/123456789/4353

Elastix. (2019). *Elastix*. Obtenido de Elastix. https.//www.elastix.org/

GitHub. (01 de 01 de 2020). *GitHub*. Obtenido de GitHub. https.//github.com/

Kumar, C. (02 de 01 de 2020). *GeekFlare.* Obtenido de GeekFlare. https.//geekflare.com/best-open-source-monitoring-software/

MariaDB. (13 de febrero de 2020). *MariaDB*. Obtenido de MariaDB. https.//mariadb.org/

Message Plus S.A. (03 de 08 de 2019). *Message Plus S.A.* Obtenido de Message Plus S.A.. http.//www.messageplus.ec/

MUNIN. (24 de 01 de 2020). *Munin org*. Obtenido de Munin org. http.//demo.munin-monitoring.org/

MySQL. (24 de febrero de 2020). *MySQL*. Obtenido de MySQL. https.//www.mysql.com/

Oscar, L. S. (2017). *Monitoreo de la infraestructura de red en la dirección distrital 18d01 salud utilizando herramientas basadas en software libre (Tesis de Postgrado).* . Universidad Regional Autónoma de los Andes, Facutad de sistemas mercantiles. Ambato. Universidad Regional Autónoma de los Andes. Recuperado el 2019, de http.//dspace.uniandes.edu.ec/handle/123456789/7414?mode=full

PHP. (07 de marzo de 2020). *PHP*. Obtenido de PHP. https.//www.php.net/

PostgresSQL. (03 de marzo de 2020). *PostgresSQL*. Obtenido de PostgresSQL. https.//www.postgresql.org/

SAS. (2019). *SAS The Power to Know*. Obtenido de SAS The Power to Know. https.//www.sas.com/es_ar/insights/analytics/big-data-analytics.html

Sergio, V. (5 de marzo de 2020). *Fundamentos sobre la gestión, monitorización y control de redes.* Obtenido de WordPress. https.//velezconde.wordpress.com/2009/06/20/fundamentos-sobre-la-gestion-monitorizacion-y-control-de-redes/

Solis, M. A. (2014). *Implemntación de plataforma de monitoreo Zabbix para sistemas de telecomunicaciones Telsur (Tesis de pregrado).* Universidad Austral de Chile, Ingeniería civil electrónica. Valdivia. Universidad Austral de Chile. Recuperado el 2019, de http.//cybertesis.uach.cl/tesis/uach/2014/bmfciu.41i/doc/bmfciu.41i.pdf

Wilman, S. P. (2014). *Propuesta de monitoreo de la infraestructura tecnológica de los servidores del ministerio de finanzas tecnológica ded los servidores del ministerio de finanzas, basado en el modelo ITIL V3 y en la herramienta HP Sitescope. (Pregrado).* Universidad Politécnica Salesiana Sede Quito, Ingeniería de Sistemas. Quito. Universidad Politécnica Salesiana. Recuperado el 2019, de https.//dspace.ups.edu.ec/handle/123456789/6822

Wong, Y. R. (2012). *Monitoreo de servicios en redes LAN (Tesis de pregrado).* Universidad Central “Marta Abreu” de Las Villas, Facultad de Matemática, Física y Computación. Santa Clara. Universidad Central “Marta Abreu” de Las Villas. Recuperado el 2019, de http.//dspace.uclv.edu.cu/bitstream/handle/123456789/7316/Tesis-Yaisel.pdf?sequence=1&isAllowed=y

Zamora, E. V. (2013). *Gestor automático de eventos en servidores mediante el uso de una matriz de escalamiento, propuesta basada en software open source (Tesis de Pregrado).* UNIVERSIDAD DE GUAYAQUIL, FACULTAD DE

CIENCIAS MATEMÁTICAS Y FÍSICAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES. GUAYAQUIL – ECUADOR. REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍA. Recuperado el 04 de 08 de 2019, de http.//repositorio.ug.edu.ec/handle/redug/2799

ANEXOS

Preguntas en la entrevista al Gerente de TI

¿Cuáles son los principales servicios que brinda la empresa?

¿Qué sistemas se priorizan en la empresa?

¿Cuál es la arquitectura de los servidores de la empresa?

¿Cómo se manejan los entornos virtuales en la empresa?

¿Cómo son brindados esos servicios por las terminales de la empresa?

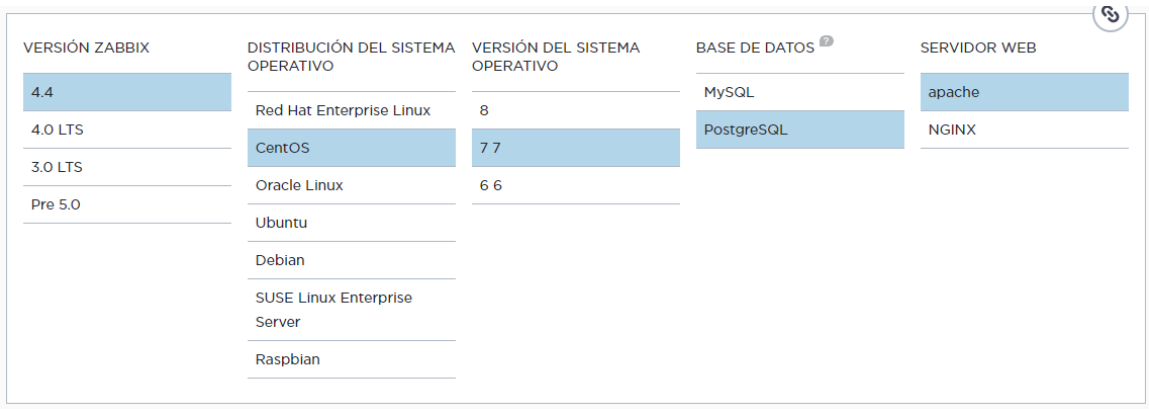
¿Qué sistemas cree que sean los que se pueden automatizar?

¿Cuáles son las principales actividades que se desean automatizar?

¿Qué servicios se pueden controlar de manera automática?

Instalación Zabbix

Para la instalación se deben establecer los parámetros del sistema en donde se va a correr Zabbix, en su página principal. https.//www.zabbix.com/download



Se procede con la descarga y actualización de paquetes bases del sistema requerido.

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.4/rhel/7/x86_64/zabbix-release-4.4-1.el7.noarch.rpm
# yum clean all
```

Ahora se instala los archivos base de Zabbix, dependiendo de si es host o servidor se pone los parámetros necesarios.

```
# yum install zabbix-server-pgsql zabbix-web-pgsql zabbix-agent
```

Una vez completado se debe levantar la base de datos, en este caso postgres, luego se inicializa con los datos iniciales.

```
# sudo -u postgres createuser --pwprompt zabbix
# sudo -u postgres createdb -O zabbix zabbix

# zcat /usr/share/doc/zabbix-server-pgsql*/create.sql.gz | sudo -u zabbix psql zabbix
```

Finalmente se editan dos directorios del árbol de Zabbix, en donde se establece la contraseña y la región.

Editar archivo /etc/zabbix/zabbix_server.conf

```
DBPassword=password
```

Edite el archivo /etc/httpd/conf.d/zabbix.conf,

```
# php_value date.timezone Europe/Riga
```

Ahora solo se inicializa y habilita los servicios.

```
# systemctl restart zabbix-server zabbix-agent httpd
# systemctl enable zabbix-server zabbix-agent httpd
```

Para acceder a la interfaz web, se debe colocar la ip del servidor seguido de Zabbix, de la siguiente manera.

```
http: // server_ip_or_name / zabbix
```

Agregar Host con cifrado

Luego de la instalación del agente de Zabbix en los agentes, estos deben conectarse al servidor, para lo cual se sigue los siguientes pasos.

Para agregar un host con cifrado primero se edita el archivo alojado en el host.

```
vim /etc/zabbix/zabbix_agentd.conf
```

Estableciendo los siguientes parámetros, (la IP definida es la del servidor)

```
Server=192.168.4.150

ListenPort=10050

ServerActive=192.168.4.150

Hostname=srva__mplus

TLSConnect=psk

TLSAccept=psk

TLSPSKIdentity=psk_10

TLSPSKFile=/etc/zabbix/zabbix_agentd.psk
```

Los archivos psk que se especifican en la configuración anterior deben ser generados con un número aleatorio de la siguiente manera.

```
openssl rand  -hex 32 >/etc/zabbix/zabbix_agentd.psk

cat /etc/zabbix/zabbix_agentd.psk
```

Se deben habilitar los servicios y puertos en el firewall, estos comandos son propios de la versión del sistema operativo.

Centos 7.

```
firewall-cmd --permanent --add-port 10050/tcp

firewall-cmd --reload
```

Y se reinicia los procesos.

Centos 7.

```
systemctl start zabbix-agent
systemctl enable zabbix-agent
```

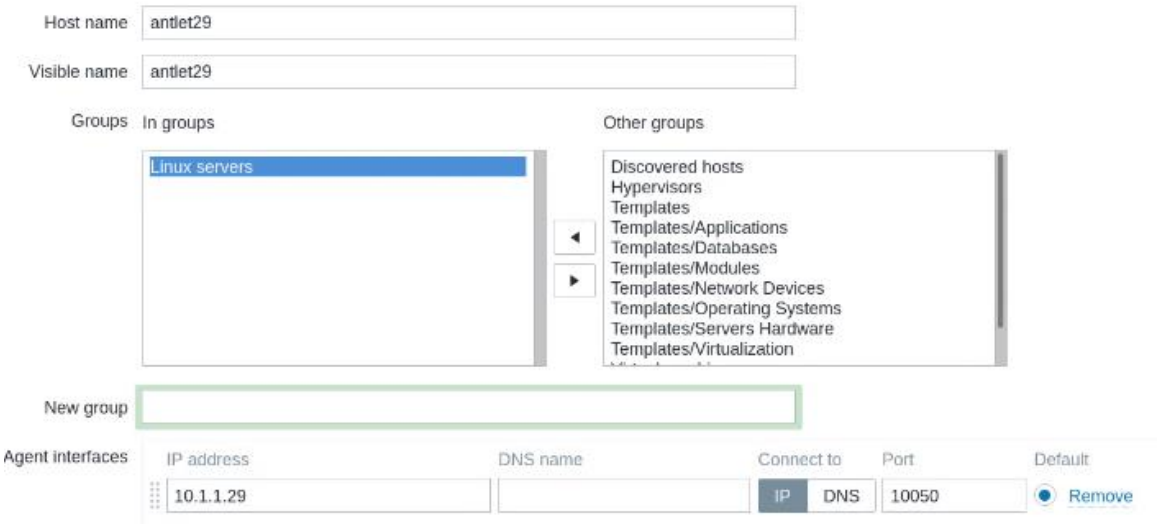
Centos 6.

```
service zabbix-agent restart
chkconfig --level 35 zabbix-agent on
```

Y se verifica el estado del sistema.

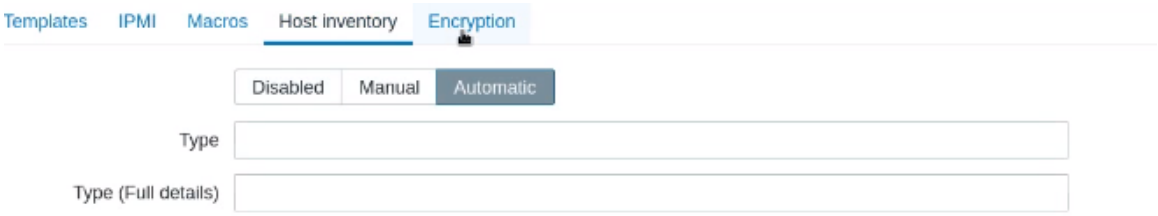
```
zabbix-agent.service - Zabbix Agent
 Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: disabled)
 Active: active (running) since Thu 2018-08-16 06:26:14 PDT; 15s ago
lain PID: 1222 (zabbix_agentd)
 CGroup: /system.slice/zabbix-agent.service
         ├─1222 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
         ├─1223 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
         ├─1224 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
         ├─1225 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
         ├─1226 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
         └─1227 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]
```

Ahora en el servidor Zabbix, se procede a agregar la configuración del host y la clave pre compartida generada en el paso anterior. Para esto se debe dirigir a la pestaña configuración y al apartado Host, aquí seleccionar nuevo host. La IP que se especifica es la del host que se está agregando.

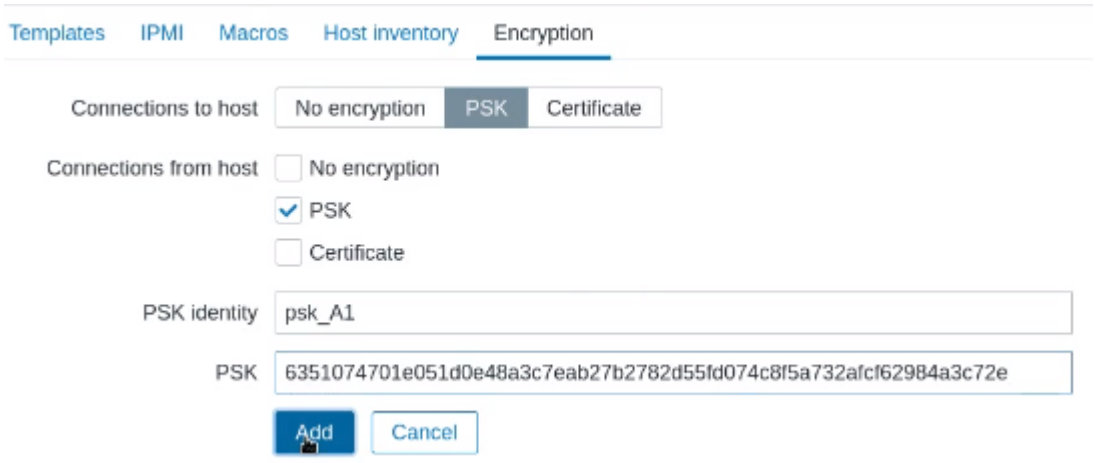


En la pestaña templates es importante agregar los siguientes parámetros.

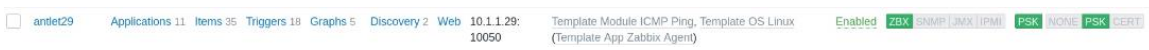
En host inventory se debe dejar en automático, a menos que se requiera especificación.



Y en el último apartado de encriptación se habilita la opción de PSK, y se establece el PSK Identity que es el nombre que se estableció en la configuración del archivo del host, y la clave PSK que es el número de 32 bits generado en ese mismo host.



Y se establece la conexión con las opciones habilitadas.



Manejo de notificaciones en Zabbix

La gestión de notificaciones en Zabbix, se hace con las Media Types, que son archivos dedicados a manejar la información generada por la plataforma. Para este paso se utiliza Pushover, el gestor que permite recibir notificaciones en el celular, y se instala de la siguiente manera.

En el terminal del servidor Zabbix, se debe instalar las dependencias que usa el sistema.

```
yum install epel-release

yum install python-pip

 pip install python-pushover

pip install --upgrade pip
```

Con Python y pushover instalado, se debe crear el script que genere las notificaciones del sistema, todo esto alojado en las alertas del sistema.

```
AlertScriptsPath=/usr/lib/zabbix/alertscripts
```

El script usado es el siguiente.

notify_pushover.py

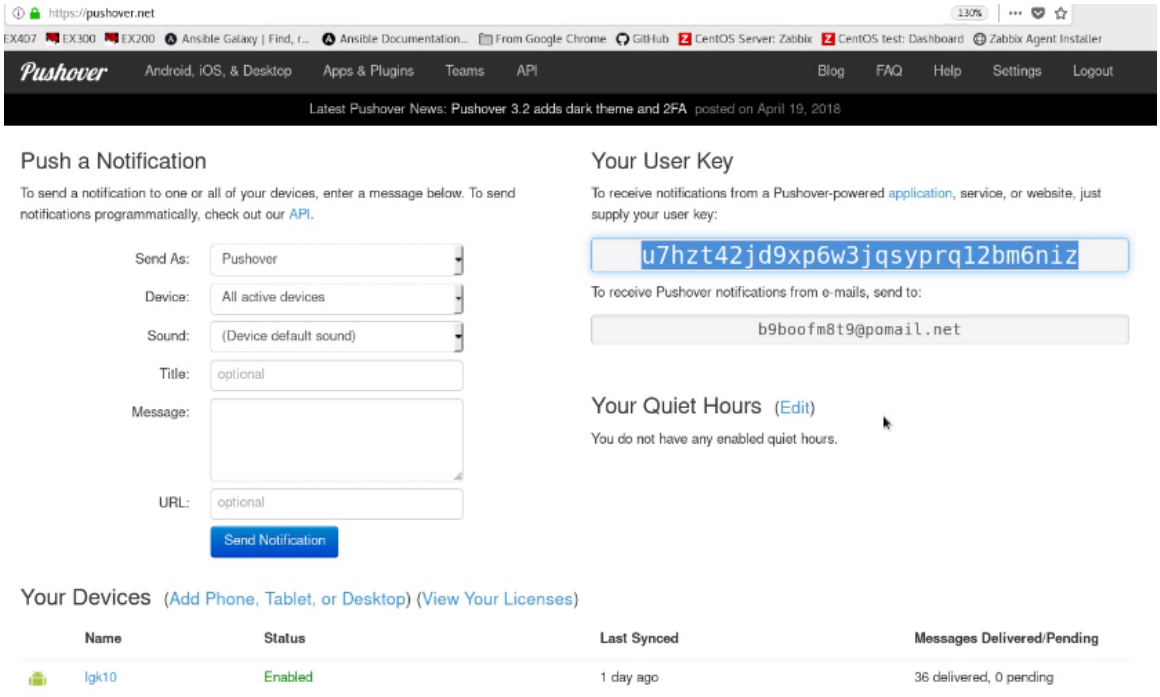
```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
# Pushover notification script for Zabbix
#
# Author:
#   Si¿½bastien RICCIO - sr@swisscenter.com
#
# Purpose:
#   Push zabbix notifications to Pushover enabled devices
#   See: https://pushover.net/
#
# Requirements:
#   pip install python-pushover
#
# Changelog:
#   20170101 - First revision
#

import argparse
import sys
import time
import pushover

#
# Settings
#
ENABLE_LOG = True
LOG_FILE = "/var/log/zabbix/notify_pushover.log"


#
# Functions
#
def l(msg):
    """
    Send log line to stdout and to LOG_FILE if logging is enabled
    """
    msg = "[%s] %s" % (logTimeStamp(), msg)

    # Print to stdout
    print(msg)

    # Output to logfile
    if ENABLE_LOG:
        try:
            lf = open(LOG_FILE, 'a')
            lf.write("%s\n" % (msg))

        except (OSError) as exc:
            print("Error while trying to log event: %s" % rlb(str(exc)))
            return False

        lf.close()
```

```
    return True


def logTimeStamp():
    """
    Return current date/time formatted for log output
    """
    return  time.strftime('%a %b %d %H:%M:%S %Y')


def rlb(thing):
  """
  Return thing with line breaks replaced by spaces
  """
  return thing.replace("\r", " ").replace("\n", " ")

#
# Main code
#

# Arguments parser
parser = argparse.ArgumentParser(description='Send Zabbix notification to Pusho
parser.add_argument('user_key_app_token', metavar=('UserKey|AppToken'), type=st
parser.add_argument('subject', metavar=('Subject'), type=str, help='Subject you
parser.add_argument('message', metavar=('Message'), type=str, help='Message you

# Argument processing
args = parser.parse_args()
user_key_app_token = args.user_key_app_token
subject = args.subject
message = args.message

# Check if UserKey and AppToken has been supplied
if "|" in user_key_app_token:
  user_key, app_token = user_key_app_token.split("|")
else:
  l("Error: you must supply both User Key and App Token separated with |")
  sys.exit(1)

# Try to login with AcessToken
try:
    po = pushover.Client(user_key, api_token=app_token)
except (pushover.UserError) as exc:
    l("Error: Can't connect to Pushover with User Key [%s]." % user_key)
    sys.exit(1)

# Try to send the notification
try:
    po.send_message(message, title=subject)
except (pushover.RequestError) as exc:
    l("Error: Can't send notification to Pushover devices: %s" % rlb(str(exc)))
    sys.exit(1)
```

Y se le dan los permisos de ejecución necesarios.

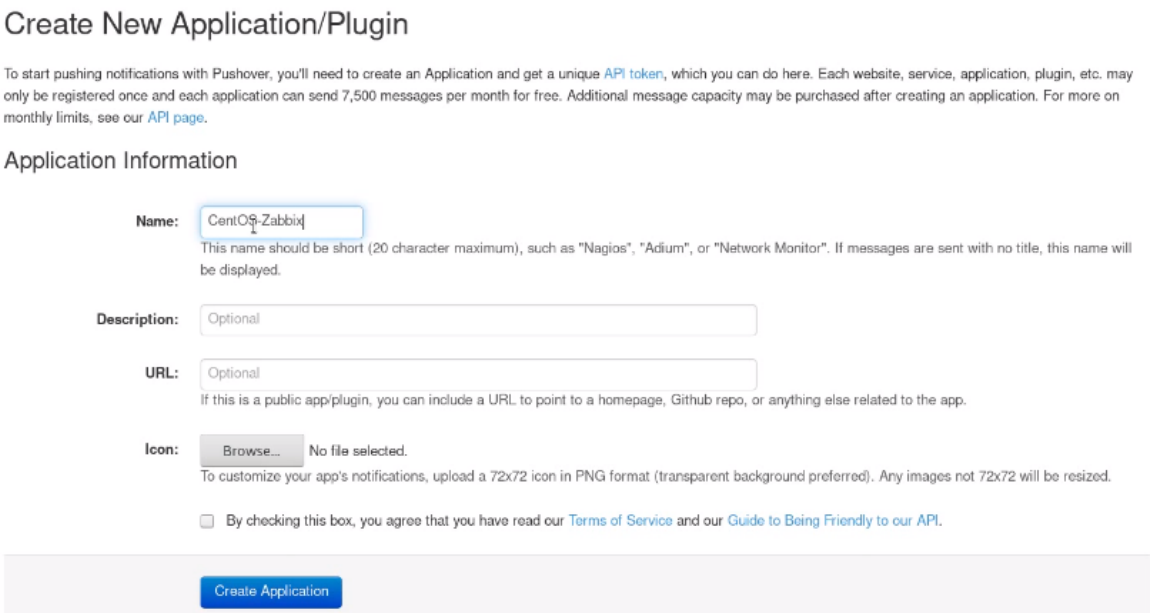
```
chmod +x notify_pushover.py
```

Ahora ya terminada la configuración del sistema, se procede a los ajustes de la aplicación, que es la encargada de la gestión de la información, para esto se debe crear una cuenta en la web https.//pushover.net/, donde se nos genera la clave de usuarios para las API's. Una vez generada la clave, se debe crear una nueva aplicación en donde se una a las notificaciones del sistema. Además, de generar la conexión con los dispositivos en donde se quiere recibir las notificaciones, esto solo descargando la app Pushover e iniciando la sesión que se creó en la web.

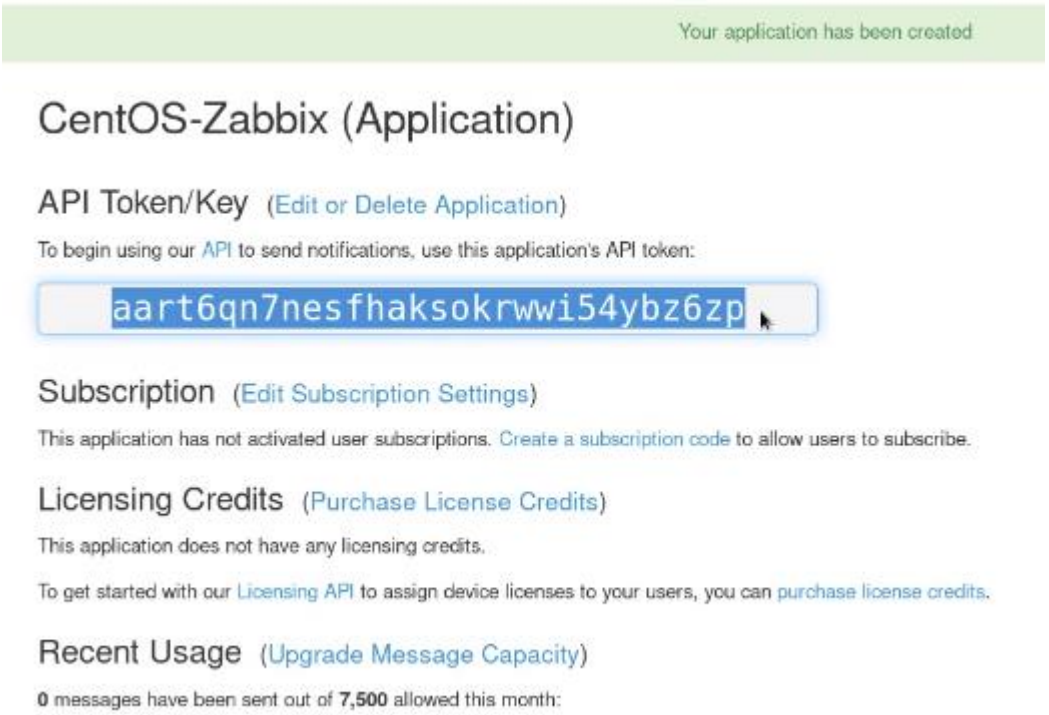


La nueva aplicación se hace en la parte de aplicaciones.

Se configura a su elección.



Se genera ahora la clave de la aplicación.

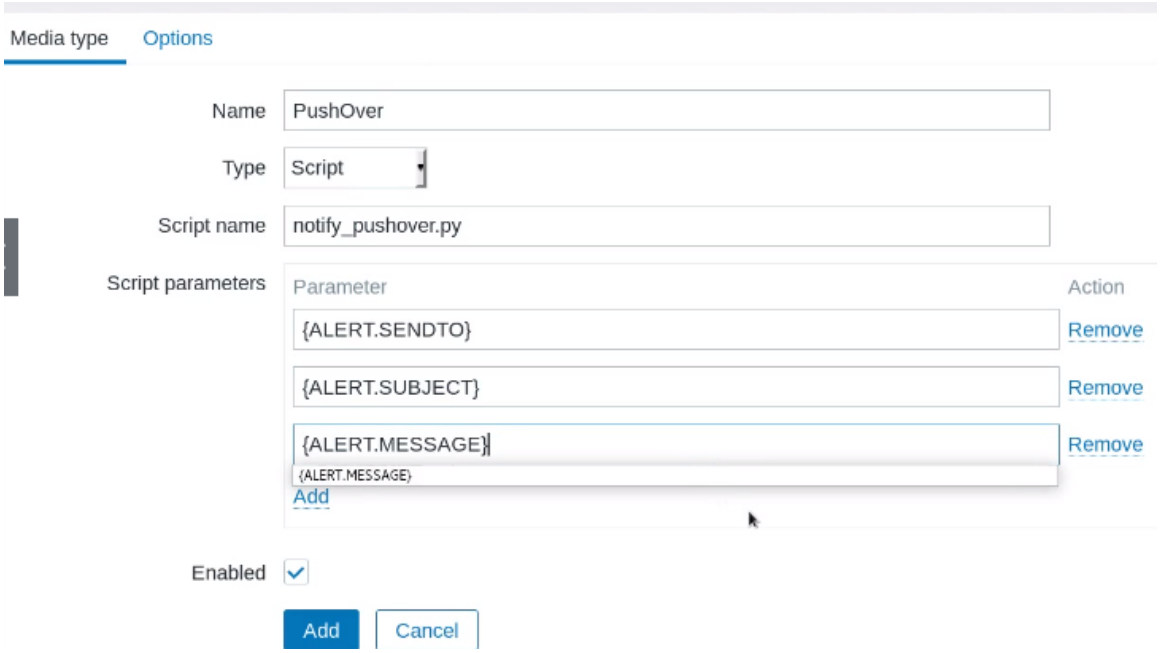


Para realizar una prueba en línea, nos dirigimos a la terminal de Zabbix, donde creamos el archivo Python con el script de notificaciones. Y lo ejecutamos con la especificación de ejecución del archivo, seguido de la clave de usuario, la clave de aplicación y los parámetros que se deben mostrar en la notificación.

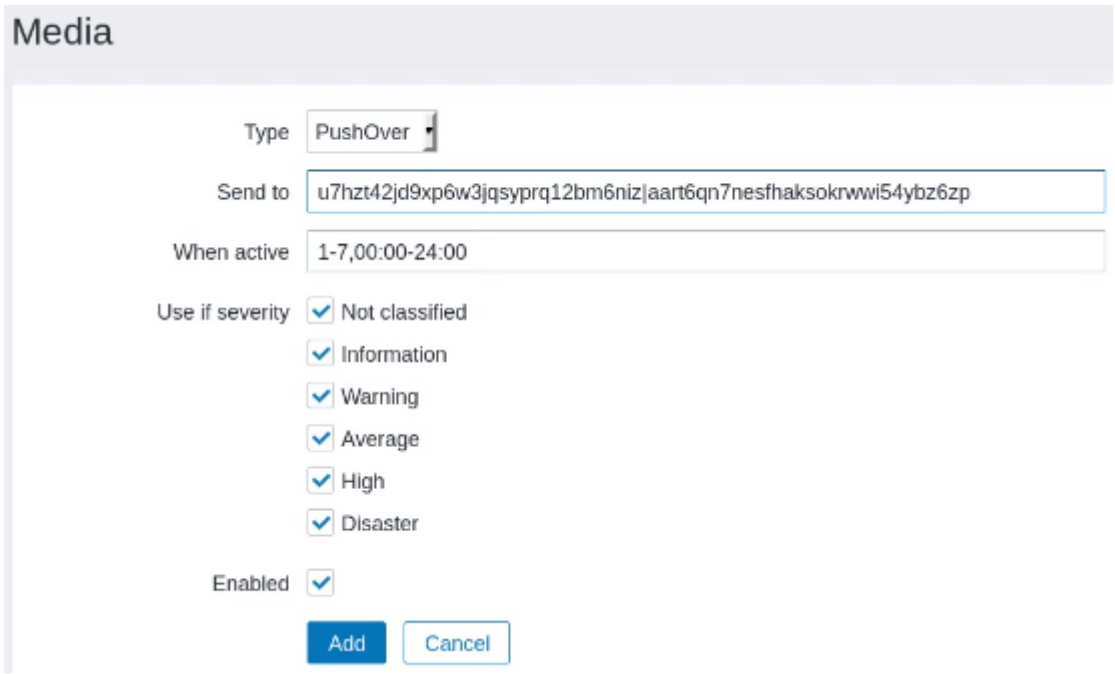
```
[root@antlet23 alertscripts]# ./notify_pushover.py 'u7hzt42jd9xp6w3jqsyprq12bm6niz|aart6qn7nesfhaksokrwwi54ybz6zp' Hola
 Probando
[Sat Aug 18 19:57:48 2018] Success: Message sent with UserKey [u7hzt42jd9xp6w3jqsyprq12bm6niz] to AppToken [aart6qn7nes
fhaksokrwwi54ybz6zp]
```

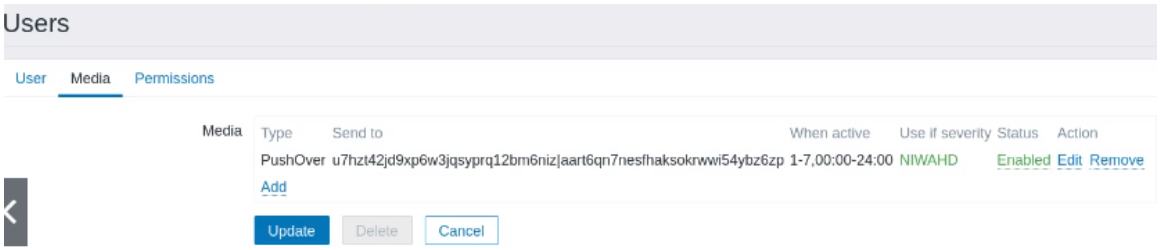
El mensaje de envió exitoso, confirma que la aplicación es capaz de enviar información desde el servidor.

Ahora la parte final consiste en conectar el script con la web de Zabbix. Para esto se debe crear en configuración en el apartado de Media Types, una nuevo, con los parámetros de Pushover, y las claves que usamos para el usuario y la aplicación.



Ahora vamos a User, y seleccionamos los usuarios que quieren recibir notificaciones, y se le agrega una nueva Media, con las configuraciones de Pushover.





Seguridad Zabbix

Para establecer un certificado que respalde la seguridad de la página web, se lo genera y firma en el mismo servidor. Para esto se comienza con la instalación de dependencias.

```
yum install  mod_ssl
```

Luego se genera el archivo de configuración con los nombres de la clave de acceso y el certificado de la encriptación.

```
[root@localhost ~]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/zabbix-ssl
.key -out /etc/pki/tls/certs/zabbix-ssl.crt
```

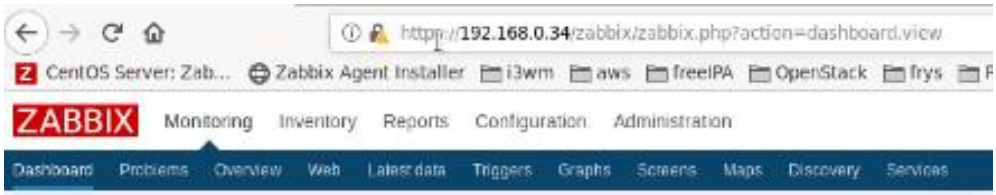
Para la certificación es posible establecer parámetros de configuración libres, al entorno gráfico y empresarial de la configuración. Para establecer parámetros debemos ir al path siguiente y hacer una copia del siguiente archivo.

```
httpd]# cd conf.d/

cp ssl.conf 00-zabbix.conf
```

Editamos esta copia, estableciendo los parámetros de inicio, de clave y de archivos de certificación. Además, se establece el bloque del puerto 80, para que no se pueda acceder a este por la interfaz web. También se puede eliminar el inicio del Apache, en la carpeta en donde se encuentra nuestro archivo de configuración, está el "welcome.conf" que se debe eliminar de la raíz, este se puede restablecer al reiniciar el servicio.

```
<VirtualHost *:80>
        ServerName zabbix
        RewriteEngine on
        RewriteRule ^(/.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

<VirtualHost _default_:443>
        ServerName zabbix
        SSLEngine on
        SSLProtocol all -SSLv2 -SSLv3
        #SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA
        SSLCipherSuite RC4-SHA:AES128-SHA:HIGH:MEDIUM:!aNULL:!MD5
        SSLHonorCipherOrder on
        SSLCertificateFile /etc/pki/tls/certs/zabbix-ssl.crt
        SSLCertificateKeyFile /etc/pki/tls/private/zabbix-ssl.key
        SSLCertificateChainFile /etc/pki/tls/certs/zabbix-ssl.crt
        DocumentRoot /usr/share/zabbix
</VirtualHost>


<Directory /usr/share/zabbix>
        Require all granted
```

Así al recargar la web, se hace la redirección al HTTPS.



Instalación Monit

Para la implementación de Monit, al ser de código libre, se inicia con la instalación de dependencias propias del sistema operativo en el que el software va a funcionar. Centos requiere de los repositorios epel-release, de la siguiente manera.

```
yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

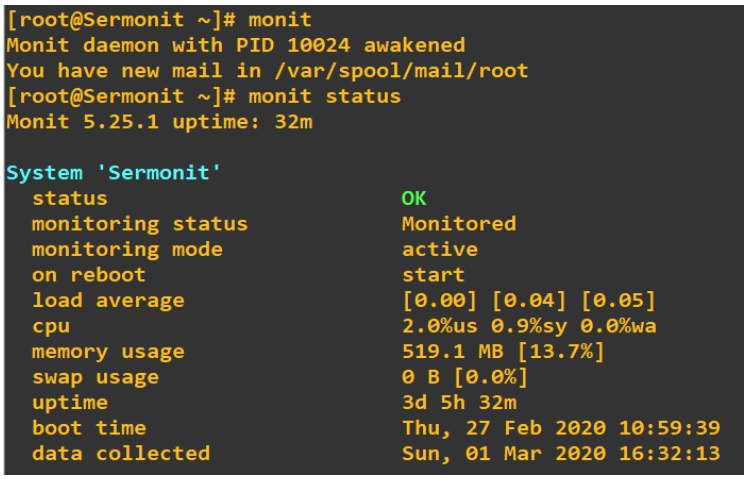
Ahora se puede instalar Monit.

```
yum -y install monit
```

Se procede a la inicialización de la dependencia, y la verificación del servicio.

```
monit
```

```
monit status
```



Con el servicio iniciado correctamente se procede a las especificaciones técnicas en el archivo de configuración, que se encuentra en el path. cd /etc/monit.conf, habilitando el demonio, el puerto, dirección y alertas de Monit, de la siguiente manera.

```
...
set daemon 60
...
set mailserver mail.mplus.ec port 587
...
set mail-format {
  from:    monit@mplus.ec
  subject: ALERTA MONIT -- $EVENT $SERVICE
  message: $EVENT: Service $SERVICE
               Fecha:        $DATE
               Accion:       $ACTION
               Host:         $HOST
               Descripcion:  $DESCRIPTION
          Saludos,
          Monit
 }
...
set alert soporte@mplus.ec
...
```

Para el apartado web, se configura en el mismo archivo de configuración, para luego habilitar el puerto que usa Monit en el firewall, esto se hace por consola, así.

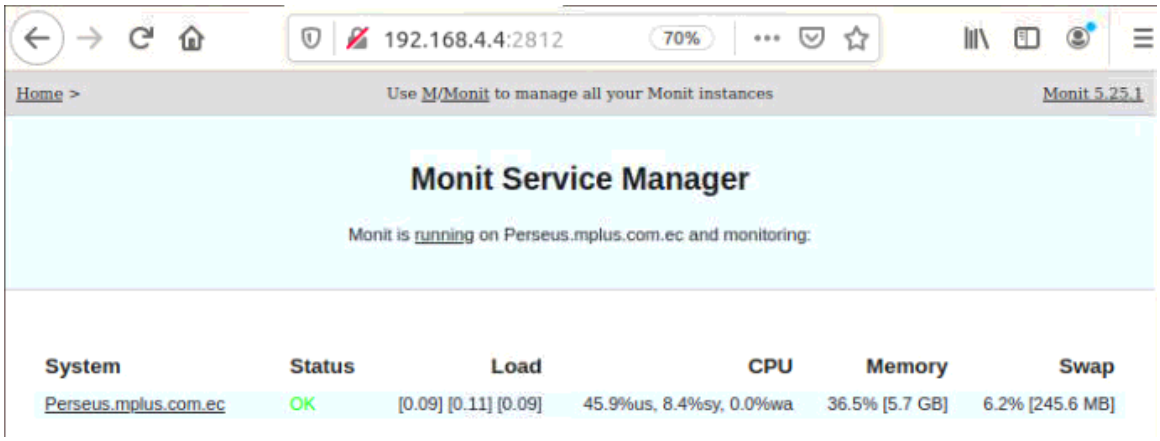
monit.conf

```
set httpd port 2812
    use address x.x.x.x      # only accept connection from localhost
    allow 0.0.0.0/0.0.0.0      # allow localhost to connect to the server and
    allow xxxx:xxxx
```

Consola.

```
systemctl restart monit
systemctl enable monit
firewall-cmd --permanent --add-port=2812/tcp
firewall-cmd --reload
```

El resultado es la interfaz de monit levantada con la monitorización del sistema en donde se aloja de manera predeterminada.



Desarrollo de los script Monit

Monit al ser un software de monitoreo y control de procesos, permite mediante su interfaz web la visualización de estos estados, pero no un control gráfico de los mismos, el control y las acciones automáticas las realiza un conjunto de los script definidos por consola, estos se pueden acoplar a un gran número de casos y

variables, para este caso se presenta el código desarrollado para el control de procesos de envío de mensajería masiva, en los siguientes.

Script de verificación; desarrollado para verificar el estado de una conexión en tiempo real mediante la verificación de su PID, en caso de no encontrar este valor, el sistema procede a ejecutar el stop, que elimina cualquier proceso inhibido, y un start de la forma en que el proceso debe ser ejecutado.

```
check process MKT1P with pidfile /home/sysmplus/GWec/var/run/smppTx_smsmarketing.pid
      stop  program = "/etc/monit.d/procesos/run_monit_MKT1 stop" with timeout 30 seconds
      start program = "/etc/monit.d/procesos/run_monit_MKT1 start"
```

El script que permite el start y stop, se desarrolla en PHP, y presenta la siguiente estructura, que solo define la manera correcta de parar e iniciar un proceso en la arquitectura de la empresa.

```
#!/bin/bash

        case $1 in
        start)
                        sudo -u sysmplus /home/sysmplus/bin/perl /home/sysmplus/GWec/bin/s
mppTxmarketing 94 smppTx_smsmarketing;
           echo "start";
           ;;
         stop)
           echo "stop";
         kill -9 $(sudo -u sysmplus cat /home/sysmplus/GWec/var/run/smppTx_smsmarketing.pi
d)
           ;;
         *)
        echo "usage: run_monit_MKT1 {start|stop}"
        ;;
esac
~
```

Debido a que los procesos de la arquitectura de la empresa son similares, se puede tomar el ejemplo anterior y redefinir con pequeños cambios a casi cualquier proceso, con un script de verificación en el sistema Monit, y un PHP que especifique las acciones con las que un proceso se debe tratar.

Notificaciones Monit

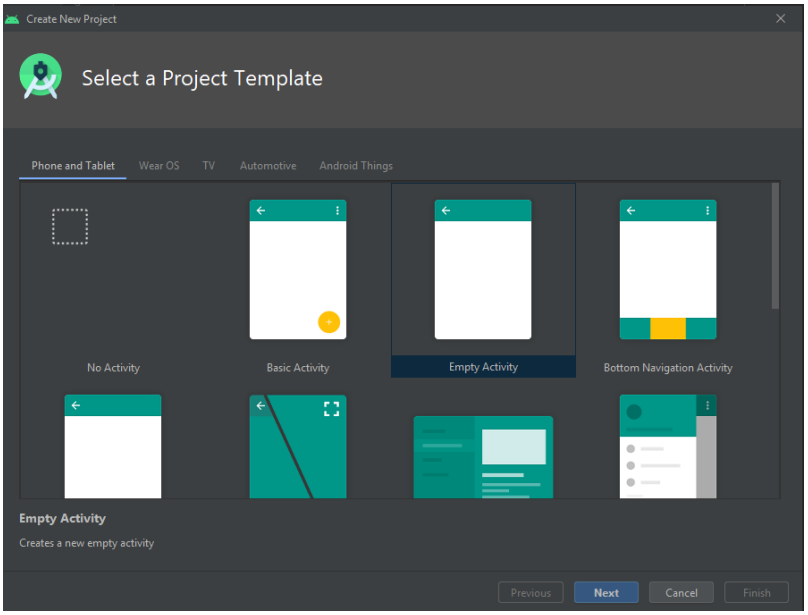
Para la configuración del correo electrónico, se debe acceder al archivo de configuración de Monit, en donde con macros se establece la forma como será notificado el correo indicado de los incidentes en los procesos. Es importante saber que el correo de monit es una propio de donde se enviaran los correos a los colaboradores especificados en set alert, aquí también se puede identificar el tipo de alertas que debe recibir cada usuario.

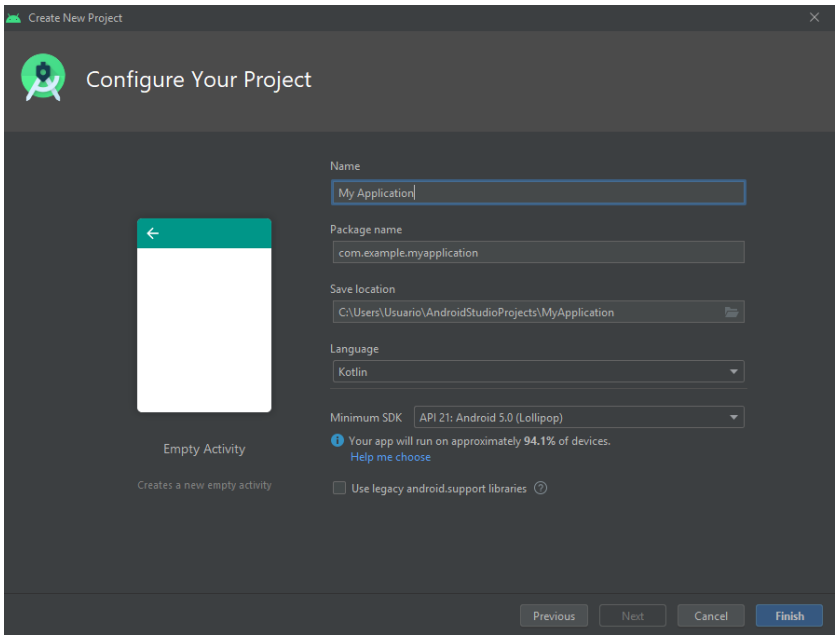
```
## Monit by default uses the following format for alerts if the mail-format
## statement is missing::
## --8<--
set mail-format {
   from:    monit@mplus.ec
   subject: ALERTA MONIT --  $EVENT $SERVICE
   message: $EVENT: Service $SERVICE
                Fecha:          $DATE
                Accion:         $ACTION
                Host:           $HOST
                Descripcion:    $DESCRIPTION

            Saludos,
            Monit
 }

#set alert lvallejo@mplus.ec
## --8<--
##
## You can override this message format or parts of it, such as subject
## or sender using the MAIL-FORMAT statement. Macros such as $DATE, etc.
## are expanded at runtime. For example, to override the sender, use:
#
# set mail-format { from: monit@foo.bar }
#
#
## You can set alert recipients whom will receive alerts if/when a
## service defined in this file has errors. Alerts may be restricted on
## events by using a filter as in the second example below.
#
 set alert lvallejo@mplus.ec                        # receive all alerts
#
```

Implementación de Aplicación

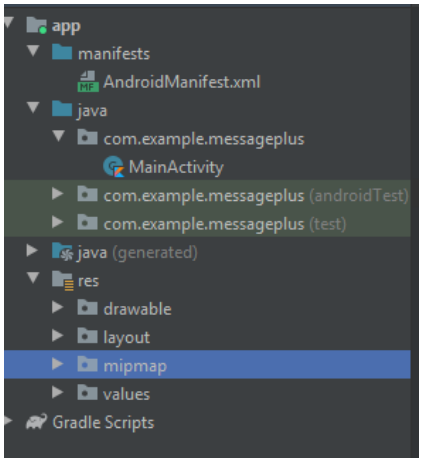
Para la implementación de la aplicación en el entorno Android, se usa Android Studio, que es un IDE de libre acceso con las librerías necesarias para la implementación de aplicación que soporten distintas versiones de este sistema operativo.

Se comienza con la realización de un nuevo proyecto, especificando el lenguaje en el que se va a basar la aplicación, el tipo de panel usado, y la API que maneja.



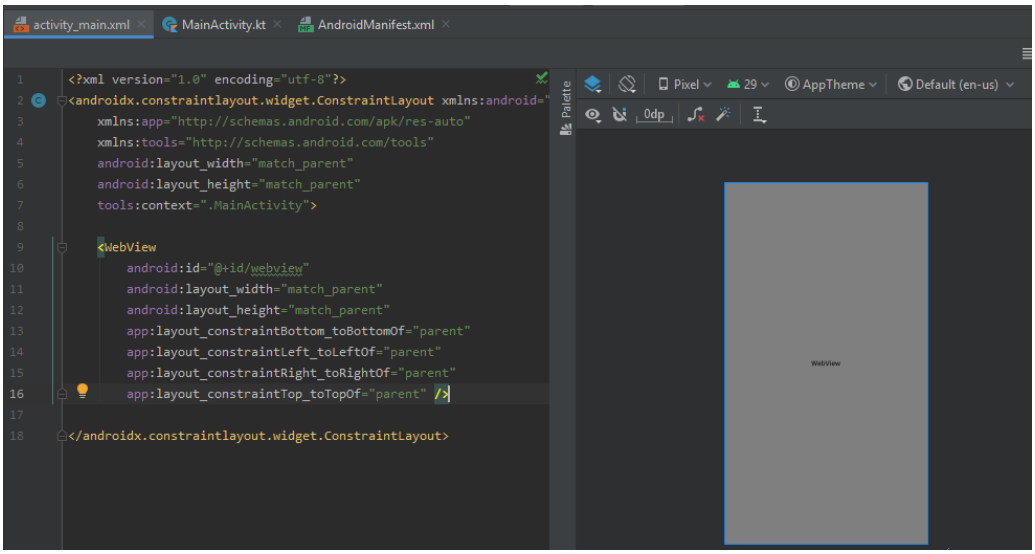


Una vez se establecen los parámetros de inicio, se crea el proyecto, y se despliega un árbol de directorios.



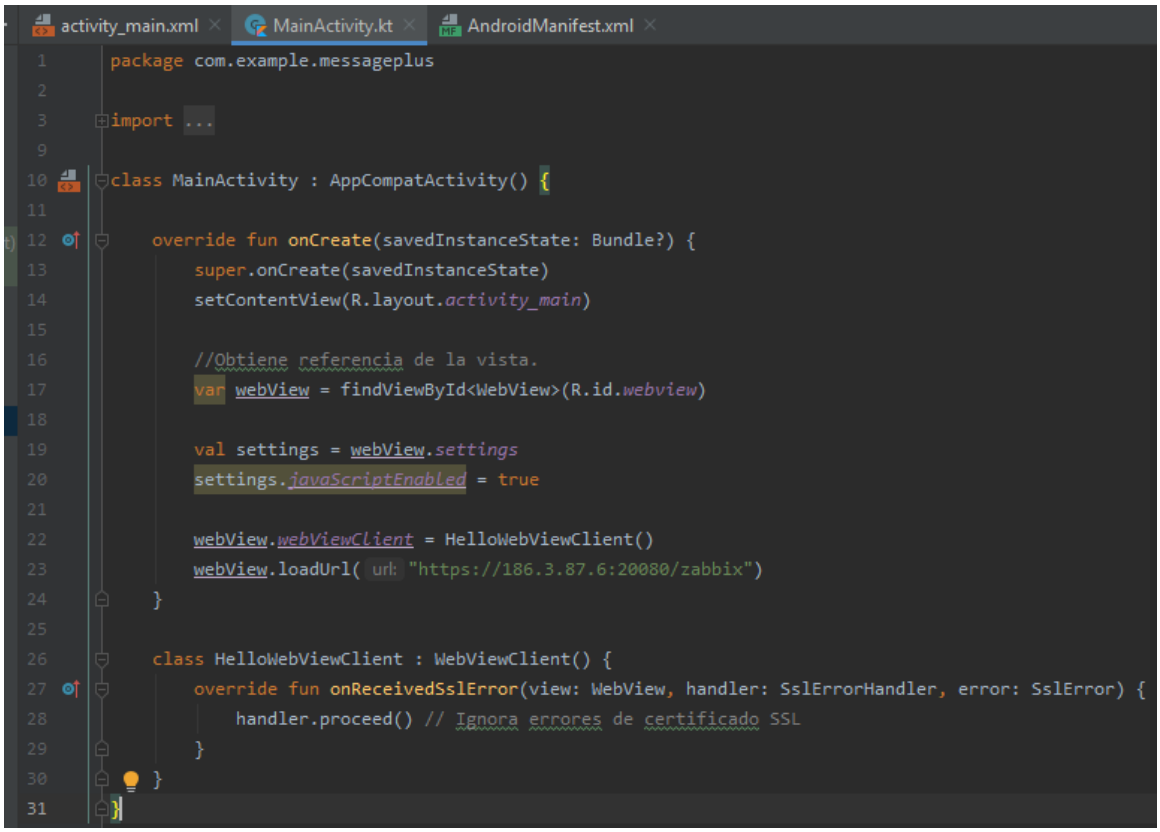
Primero el activity_main.xml, que maneja la parte que se muestra en la aplicación, luego el MainActivity.kt, que se encarga de la parte lógica de la aplicación. Además, en el apartado app/manifests/ se encuentra AndroidManifest.xml, que se encarga del entorno de compilación, que es donde se especifican los parámetros de entorno para la variable, a continuación, se describe el código de cada apartado.

activity_main.xml. define solo el contenedor web que va a alojar la definición de la plataforma, ocupando la mayor parte de la pantalla del dispositivo.



MainActivity.kt. Establece la parte lógica, definiendo las variables que guardan el contenedor web, los ajustes de JavaScript, y la normalización de la dirección web.

También se define un método de cliente web para el manejo de los certificados de la página, debido a que si esto se presenta no permite el ingreso directo al contenido de la misma, como lo haría en un navegador web.

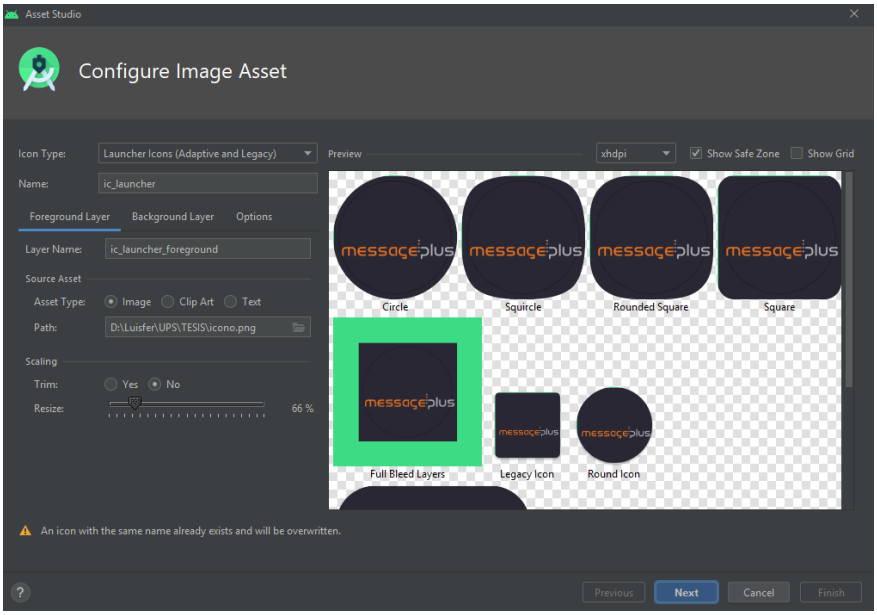


AndroidManifest.xml. Aquí se define los accesos de la aplicación en el dispositivo, lo más importante es que debe acceder a la red que usa el dispositivo.

activity_main.xml | MainActivity.kt | AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.messageplus">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="MessagePlus"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Para la finalización de la aplicación se gestiona la exportación del APK y manejo de logo de la empresa. El ícono se agrega dando clic izquierdo en la carpeta res/mipmap/ y seleccionando. New/ Image Asset, donde se puede definir logo y colores del ícono que se mostrara en el dispositivo. Se puede generar una gran cantidad de íconos que se adapten a múltiples sistemas, así como definición de cabecera, fuente y color.



Para la exportación del APK, se usa la pestaña general Build, luego el apartado Build Bundels(s) APK(s) y Build APK(s), al finalizar el proceso Android Studio nos redirigirá al luegar en donde se genera el archivo necesario.