

Tipo de artículo: Artículo original
Temática: Software Libre
Recibido: 30/06/2021 | Aceptado: 01/10/2021

Monitoreo pasivo en Nova Servidores a través de una aplicación móvil

Passive Monitoring in Nova Servers through a mobile application

Hanny Valdés Hernández ^{1*} <https://orcid.org/0000-0002-9937-1340>

David Hiram Hernández Peña ² <https://orcid.org/0000-0002-5493-0532>

¹ Universidad de las Ciencias Informáticas, Facultad 1, Centro de Software Libre. Carretera San Antonio de los Baños km 2 ½ Finca Torrens, municipio La Lisa, La Habana, Cuba. CP: 19370. {[hanny](mailto:hanny@uci.cu), [dhpenna](mailto:dhpenna@uci.cu)}@uci.cu

*Autor para la correspondencia (hanny@uci.cu)

RESUMEN

Como parte del proceso de desarrollo de software y la migración nacional a software libre surge Nova Servidores, variante de la distribución cubana GNU/Linux Nova desarrollada en el Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI). Dicha variante está orientada a los entornos de las empresas cubanas usando estándares abiertos y utiliza la herramienta Zabbix para el monitoreo y control de los recursos, aplicaciones y servicios de los servidores. La presente investigación tiene como objetivo desarrollar una aplicación móvil para el monitoreo pasivo de servidores, como una tarea imprescindible para que el cliente/usuario tenga la certeza, al saber que están encargándose de velar por el correcto funcionamiento de sus servidores, sin importar el horario o la fecha. Esta aplicación fue sometida a pruebas de software para verificar su calidad y correcto funcionamiento. Al concluir la investigación se

obtuvo una aplicación móvil para el monitoreo pasivo de Nova Servidores, que permite alertar a los administradores de red ante fallas en los servidores y visualizar el estado de los recursos y servicios.

Palabras clave: aplicación móvil; estándares abiertos; monitoreo pasivo; Nova Servidores; Zabbix.

ABSTRACT

As part of the software development process and the national migration to free software, Nova Servidores arises, a variant of the Cuban GNU / Linux Nova distribution developed at the Free Software Center (CESOL) of the University of Informatics Sciences (UCI). This variant is oriented to the environments of Cuban companies using open standards and uses the Zabbix tool to monitor and control the resources, applications and services of the servers. The present research aims to develop a mobile application for passive monitoring of servers, as an essential task so that the client / user has the certainty, knowing that they are in charge of ensuring the correct functioning of their servers, regardless of the schedule. or the date. This application was subjected to software tests to verify its quality and correct operation. At the conclusion of the investigation, a mobile application was obtained for passive monitoring of Nova Servidores, which allows to alert network administrators to server failures and view the status of resources and services.

Keywords: mobile application; open standards; passive monitoring; Nova Servidores, Zabbix.

Introducción

Gracias a las Tecnologías de la Información y las Comunicaciones (TIC), las organizaciones han conseguido obtener importantes beneficios, entre los que caben mencionar la mejora de sus operaciones, llegada a una mayor cantidad de clientes, la optimización de sus recursos, la apertura a nuevos mercados, un conocimiento más profundo acerca de las necesidades de la clientela para brindarles un servicio de mejor calidad y una comunicación más fluida, no sólo con sus empleados sino también con sus clientes y

proveedores. En pocas palabras, las TIC les permiten lograr aumentar considerablemente su eficiencia (Aniel, 2013).

En general empresas con un amplio número de miembros adoptan como práctica ofrecerle servicios a los mismos. Algunos de los servicios son correo, mensajería instantánea y publicación de información utilizando servidores centrales. En la actualidad existen dos factores que han dado lugar a un aumento en el número de servidores a administrar, el crecimiento de la información almacenada de manera virtual y del uso masivo de la computación. Incluso en pequeñas y medianas empresas, se puede encontrar una persona que administre entre uno o más servidores, números que podrían aumentar en dependencia de la cantidad de servicios que se ofrezcan.

No conocer la información acerca del tráfico que atraviesa la red, ¿qué enlace está saturando el ancho de banda o qué servicio está haciendo que la carga de los servidores sea elevada? hace imposible tener una red de telecomunicaciones óptima ya que en cualquier momento los servidores pueden caerse y detener servicios de vital importancia para la comunicación de la empresa. Por lo que es de gran importancia el monitoreo constante y preciso que permita aprovechar al máximo los recursos de hardware, prevenir incidencias y detectar los problemas cuanto antes, además de ahorrar costes y tiempo, monitoreo tanto con software automático como también mediante chequeos manuales.

El monitoreo de red de servidores de una empresa, permite analizar y determinar posibles fallas o problemas que estén presentando estos y así informar a los administradores de redes para generar alarmas de su estado. Esta necesidad obliga a adecuar un software mejorado y poderlo adecuar a los sistemas de la empresa y así prevenir problemas que se puedan presentar a futuro, debido a que cuando se presentan retrasos o fallas se generan pérdidas económicas y pérdidas de tiempo para las empresas, por tal razón hoy en día se preocupa más por asegurar y mantener la disponibilidad y rendimiento de la red para que así se eviten demoras en los tiempos de respuesta y uso del ancho de banda. (García, 2020)

Existen una gran cantidad de herramientas para realizar el monitoreo, estas pueden ser sistemas web, aplicaciones de escritorio y aplicaciones móviles. Estas últimas muy utilizadas en la actualidad debido a su portabilidad que brindan los dispositivos móviles y el intercambio constante de estos dispositivos con las personas.

Cuba, en aras de ganar en soberanía tecnológica, así como garantizar la informatización de todas las esferas de la sociedad, se apoya en el Centro de Software Libre de la UCI, cuyo objetivo es el desarrollo de la distribución cubana de GNU/Linux Nova. La selección de Nova para la migración del país responde a las necesidades de la informatización segura de la sociedad cubana usando estándares abiertos y adaptada a los entornos de las empresas cubanas. La variante para servidores cuenta con la configuración fácil e intuitiva a través de la herramienta nova-manager destinada a la administración de los servicios telemáticos y la compatibilidad con hardware obsoleto en el entorno empresarial (Nova, 2016). CESOL ha desempeñado un papel fundamental en el proceso de migración debido que sus trabajadores son los encargados de desarrollar el mismo. Para ello realizan un diagnóstico que permite determinar el estado tecnológico en las instituciones, luego proceden a la instalación de Nova y sus herramientas, entre ellas Zabbix, la cual garantiza seguimiento de los recursos y servicios de los servidores.

En las empresas cubanas donde los especialistas de CESOL han desplegado la distribución Nova Servidores, se ha detectado que la distribución de los servidores está dispersa en diferentes áreas de la entidad y algunas cuentan con pocos administradores de red. En empresas de gran infraestructura existe personal calificado encargado de controlar el estado los servidores y la seguridad informática, además un grupo de soporte técnico que se encarga de las tareas de mantenimiento y control de los activos físicos tecnológicos, cableado de red, computadoras, etc. En entidades de baja infraestructura el administrador de red cumple las funciones que le corresponden y también se desempeña como soporte técnico por lo que no puede estar pendiente al estado de los servidores a tiempo completo.

Actualmente los administradores que utilizan Zabbix para realizar el monitoreo de los recursos y servicios en servidores, han demostrado que si no se encuentran físicamente en la estación de trabajo no pueden detectar la ocurrencia de fallas de algunos de estos servicios ya que no cuentan con una herramienta que les permita el monitoreo remoto de sus servidores. Esta situación provoca falta de control de los recursos de los servidores, falta de disponibilidad de servicios e información y demora en la solución de problemas relacionados con los servidores. Lo antes planteado ocasiona afectaciones directas en los procesos fundamentales de las instituciones lo que trae consigo pérdidas de tiempo y baja productividad de las mismas. Con vista a dar respuesta a la problemática planteada, el objetivo general está dirigido a desarrollar una aplicación móvil para el monitoreo pasivo de Nova Servidores.

Métodos o Metodología Computacional

Para el desarrollo de la investigación el objeto de estudio está orientado al proceso de monitoreo pasivo en servidores, delimitándolo al proceso de monitoreo pasivo en Nova Servidores desde dispositivos móviles. Se utilizaron los métodos científicos y técnicas que a continuación se mencionan: Analítico-Sintético, Inductivo-Deductivo, Entrevista y Observación.

El monitoreo es el proceso de mantener la vigilancia sobre la existencia y magnitud de cambio de estado y flujo de datos en un sistema informático. Tiene como objetivo identificar fallas y ayudar en su posterior eliminación. Las técnicas utilizadas en el control de la información de los sistemas informáticos se cruzan con los campos de procesamiento en tiempo real, estadísticas y análisis de datos. Un conjunto de componentes de software utilizados para la recopilación de datos, su procesamiento y presentación es llamado un sistema de monitoreo (Pérez, 2013).

El monitoreo de red es un sistema que realiza el un control constante de una red de ordenadores, intentando detectar defectos y anomalías; en caso de encontrar algún desperfecto, envía un informe a los administradores (Ana Gardey, Julián Pérez Porto, 2013).

El enfoque monitoreo pasivo se basa en la obtención de datos a partir de recolectar y analizar el tráfico que circula por la red. Se emplean diversos dispositivos como *sniffers*^a, ruteadores, computadoras con software de análisis de tráfico y en general dispositivos con soporte para SNMP^b, RMON^c y Netflow^d (Romero, 2018).

Técnicas de monitoreo pasivo

1. Solicitudes remotas: Mediante SNMP: Esta técnica es utilizada para obtener estadísticas sobre la utilización de ancho de banda en los dispositivos de red, para ello se requiere tener acceso a dichos

dispositivos. Al mismo tiempo, este protocolo genera paquetes llamados *traps* que indican que un evento inusual se ha producido.

2. Otros métodos de acceso: Se pueden realizar scripts^e que tengan acceso a dispositivos remotos para obtener información importante a monitorear.

Captura de tráfico

Se puede llevar a cabo de dos formas:

1. Mediante la configuración de un puerto espejo en un dispositivo de red, el cual hará una copia del tráfico que se recibe en un puerto hacia otro donde estará conectado el equipo que realizará la captura.
2. Mediante la instalación de un dispositivo intermedio que capture el tráfico, el cual puede ser una computadora con el software de captura o un dispositivo extra. Esta técnica es utilizada para contabilizar el tráfico que circula por la red.

Análisis del tráfico

Se utiliza para caracterizar el tráfico de red, es decir, para identificar el tipo de aplicaciones que son más utilizadas. Se puede implementar haciendo uso de dispositivos *probe* que envíen información mediante RMON o a través de un dispositivo intermedio con una aplicación capaz de clasificar el tráfico por aplicación, direcciones IP origen y destino, puertos origen y destino.

Flujos

También utilizado para identificar el tipo de tráfico utilizado en la red. Un flujo es un conjunto de paquetes con: o La misma dirección o El mismo puerto TCP origen y destino o El mismo tipo de aplicación. Los flujos pueden ser obtenidos de ruteadores o mediante dispositivos que sean capaces de capturar tráfico y transformarlo en flujos. También es usado para tareas de facturación.

Estrategias de monitoreo

Antes de implementar un esquema de monitoreo se deben tomar en cuenta los elementos que se van a monitorear, así como las herramientas que se utilizarán para esta tarea.

¿Qué monitorear? Una consideración muy importante es delimitar el espectro sobre el cual se va a trabajar. Existen muchos aspectos que pueden ser monitoreados, los más comunes son los siguientes: utilización de ancho de banda, consumo de CPU^f, consumo de memoria, estado físico de las conexiones, tipo de tráfico, alarmas y servicios (Web, correo, bases de datos, proxy).

¿Qué son las alertas? Es la capacidad de un sistema de monitoreo para detectar y notificar a los operadores sobre eventos significativos que denotan un grave cambio de estado. La notificación se conoce como alerta y es un mensaje simple que puede adoptar múltiples formas: correo electrónico, SMS^g, mensaje instantáneo o una llamada telefónica. La alerta se transmite al destinatario apropiado, es decir, una parte responsable de tratar el evento. La alerta a menudo se registra en forma de boleto en un Sistema de Seguimiento de Problemas (Ligus, 2013).

La eficiencia en el proceso de monitoreo implica el conocimiento de las técnicas de monitoreo pasivo que existen, la captura y análisis del tráfico en la red, los flujos y la estrategia a utilizar. Por tanto, se hace necesario enfocar el análisis a aquellas aplicaciones desarrolladas para Zabbix, que es la herramienta utilizada por Nova Servidores para realizar el monitoreo del sistema.

Durante todo el proceso se identificaron 25 soluciones informáticas, todas fueron aplicaciones móviles, 14 de estas soluciones no contaban con código fuente disponible y muy poca documentación por lo que se desechan de la investigación, otras 6 eran de carácter privativo por lo que no se tenía acceso ni a documentación ni a código. En resumen, se determinaron para la investigación 5 aplicaciones móviles de las cuales se tienen en cuenta las siguientes características: uso de la API de Zabbix, soporte para zabbix, tipo de autenticación, disponibilidad de código fuente, funcionamiento fuera de línea y plataformas.

Tabla 1 - Comparación entre las herramientas analizadas.

	ZAX Zabbix Systems Monitoring	ZabbixAlert	Andzabbix	Zbx. Zabbix Monitoring Client	Zabbkit
Soporte para zabbix	2x y 3	2.0 o superior	2.0.x	2.0.x	3.0
Funcionamiento fuera de línea	No	No	Si	No	No
Uso api de Zabbix	No	No	No	Si	No
Disponibilidad de código fuente	No	Si	No	No	Si
Plataformas	Android y iOS	Android	Android	Android	Android y iOS
Tipo de autenticación	HTTP	Soporta HTTP y HTTPS	Básica	htaccess	Soporta HTTP y HTTPS

Una vez realizado el análisis comparativo, se puede concluir que: son herramientas que han sido desarrolladas por programadores independientes para entornos específicos, Zbx. Zabbix Monitoring Client y Zabbkit están disponibles para android 4.4 o superior, las restantes no soportan las versiones superiores a 4.2 y Andzabbix y Zbx. Zabbix Monitoring Client no tienen soporte para zabbix superior a 2.0.x. Por tanto, se hace necesario realizar una aplicación que responda a las dificultades expuestas en la problemática, además de contribuir al desarrollo de aplicaciones propias y eliminar la dependencia de software externo.

Se extraen de las aplicaciones estudiadas funcionalidades claves para el desarrollo de la solución como son: visualización de recursos, tales como, carga de CPU, RAM, uso de disco, actividad de red y envío de notificaciones.

Resultados y discusión

Actualmente Nova Servidores cuenta con la herramienta Zabbix encargada del monitoreo. Esta herramienta cuenta con una interfaz de comunicación, de donde obtiene la información de los recursos que este posee (CPU, RAM, discos duros, tráfico de la red, servicios), además genera alertas de monitoreo de servicios y recursos. Para que los administradores de red logren un control total de la herramienta deben estar

físicamente en la estación de trabajo, lo cual es una limitante ya que estos cumplen otras actividades y tareas dentro de la institución.

Para obtener una representación de la solución se define la arquitectura N-Capas basada en el patrón arquitectónico Modelo-Vista-Controlador (MVC). El patrón arquitectónico utilizado por Ionic es Modelo-Vista-Controlador (MVC), en la Figura 1 se muestra la representación de este patrón en el componente Mostrar uso recursos.

El patrón MVC surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. Es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento. A partir del uso de *frameworks* basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores (González, et al., 2012).

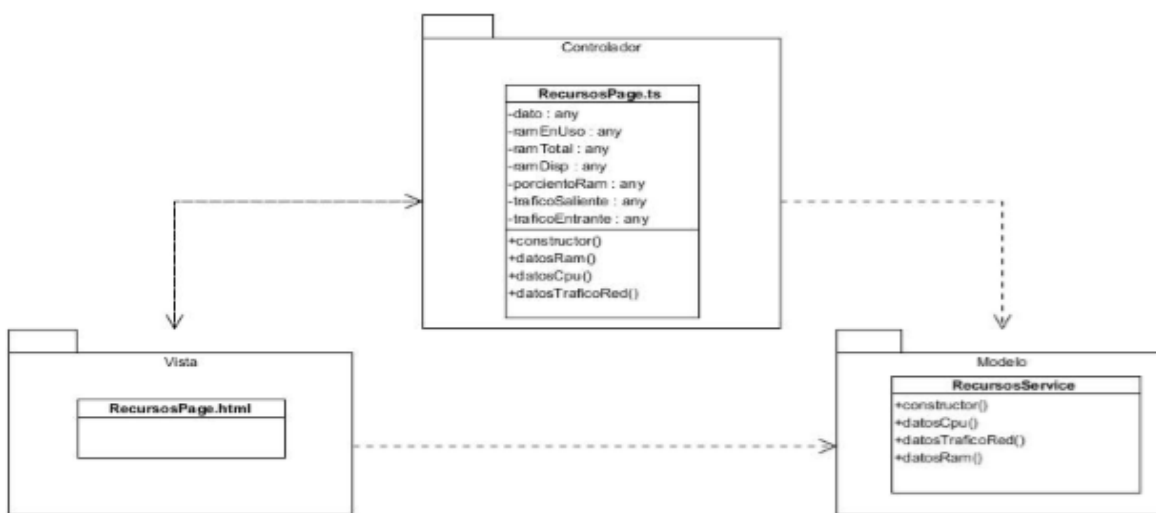


Fig. 1 - Diagrama de paquetes aplicando el patrón MVC en el componente "Mostrar uso de recursos".

El **Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. En la aplicación desarrollada el modelo se evidencia en las clases empleadas para almacenar la información proveniente del api de zabbix en forma de objetos para su posterior utilización.

La **Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. En este caso las vistas serían las plantillas empleadas para la representación visual de la aplicación. Las plantillas en el *framework Ionic* son almacenadas con la extensión *html*. Las plantillas constituyen una combinación de código *html* y un conjunto de directivas provenientes del *framework Angular*.

El **Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. En la solución propuesta los controladores son clases escritas en *TypeScript*, encargados de vincular las plantillas con los modelos y gestionar las peticiones de los usuarios.

La organización lógica de este sistema se aprecia a través de los componentes y las relaciones de dependencias entre ellos, como se muestra en la siguiente figura.

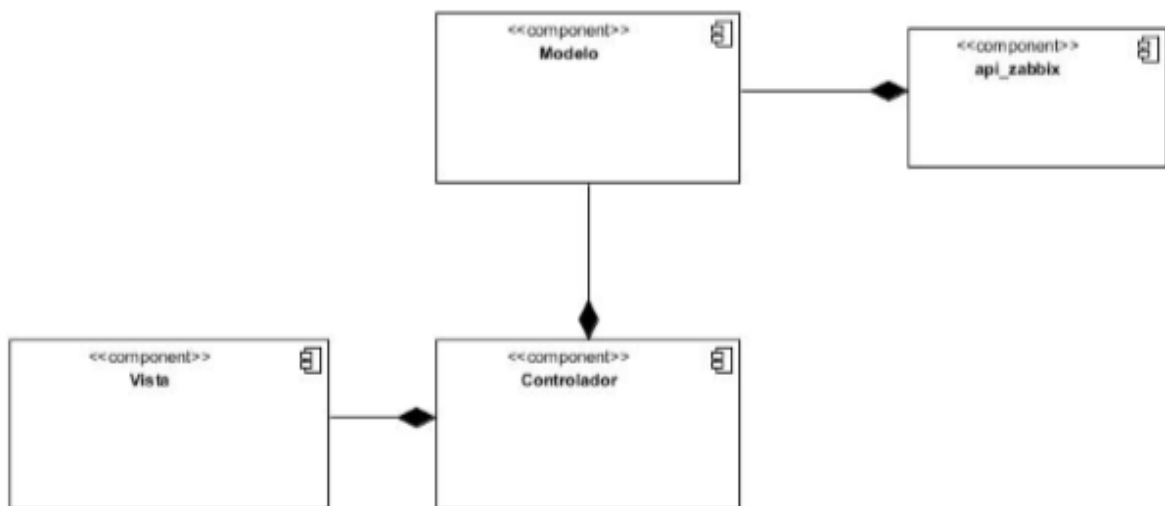


Fig. 2 - Diagrama de componentes de la aplicación.

En la siguiente figura se muestra la disposición física de los componentes que integran el sistema, describiendo el despliegue físico de información generada por el programa de *software* en los componentes de *hardware*.

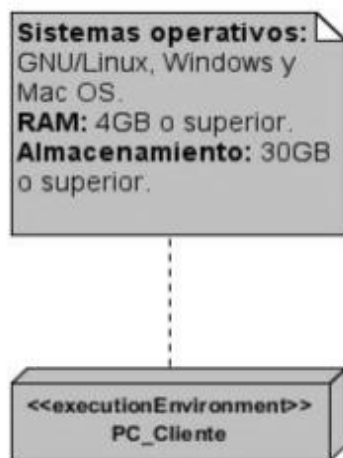


Fig. 3 - Diagrama de despliegue.

La utilización del *framework Ionic* requiere la instalación de varias aplicaciones o utilidades de consola en el ambiente de desarrollo. Los pasos a seguir para crear proyectos *Ionic* en el entorno de trabajo son (ionicframework, 2017):

1. Instalar NodeJS^h en correspondencia con el sistema operativo utilizado.
2. Instalar una versión de Ionic CLI y Cordova. El comando que posibilita la instalación de ambas herramientas de manera global es: `npm install -g ionic cordova`.
3. Descargar una aplicación básica de ejemplo para iniciar el proceso de desarrollo. El comando `ionic start [parámetro 1] [parámetro 2]` posibilita la descarga de las aplicaciones predeterminadas. El primer parámetro se refiere al nombre de la aplicación, mientras que el segundo lo hace al tipo de aplicación a descargar. Existen tres tipos básicos: *blank* (para una estructura básica y en blanco), *tabs* (con la navegación por *tabs* incluida) o *sidemenu* (para incluir un menú lateral). Un ejemplo con el tipo *sidemenu* es: *ionic start MiAplicacion sidemenu*.

4. Visualizar el resultado en el navegador. Para ello Ionic proporciona el siguiente comando: *ionic serve*. Esta acción crea un servidor web que muestra el resultado de la ejecución de la aplicación en el navegador predeterminado. Mediante las herramientas de emulación es posible comprobar, en algunos navegadores, la apariencia de la aplicación en los sistemas operativos móviles iPhone y Android. Otra de las facilidades de desarrollo brindadas por el *framework* es la recarga automática al realizar alguna modificación en los ficheros del proyecto.
5. Desplegar en el dispositivo la aplicación. Para el despliegue en Android se debe ejecutar el comando: *ionic platform add Android* que permite generar el código del proyecto para la plataforma Android. Este comando incluye en el proyecto las librerías necesarias en el lenguaje Java y los recursos propios de una aplicación nativa.
6. Realizar la construcción (*build*) y compilación (*run*) del proyecto. Para cada uno de estos pasos Ionic brinda un comando específico. Sin embargo, el segundo incluye al primero, así que solamente es necesario ejecutar: *ionic run Android*.

Pruebas de software

La aplicación obtenida como cualquier sistema de software que se desea desarrollar con calidad debe ser probado correctamente.

La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación. Una vez generado el código fuente es necesario probar el software para descubrir y corregir la mayor cantidad de errores posibles antes de ser entregado. Su objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Estas se dividen principalmente en los siguientes niveles: pruebas de unidad, de integración, de validación, de sistema y de aceptación (Pressman, 2011).

Conclusiones

Durante la presente investigación se realizó un análisis sobre el proceso de monitoreo pasivo de servidores posibilitando la comprensión de sus características y funcionamiento. Se enfatizó en el estudio de las características de aplicaciones móviles para este tipo de monitoreo, lo que posibilitó obtener funcionalidades para el desarrollo de la solución.

La solución informática de la presente investigación se sustenta en tener una aplicación móvil que permita el monitoreo pasivo de Nova Servidores a partir del resultado obtenido con el estudio de herramientas homólogas y garantiza que el administrador pueda estar en otras áreas de la institución, incluso fuera de esta, y seguir recibiendo alertas del monitoreo, observar consumo de recursos y estados de servicios.

El uso de software de código abierto y de múltiples plataformas nos permite brindar no solo centralización, confort, flexibilidad y monitorización a los servicios de la empresa, sino una solución que cumple con las normativas legales vigentes en el país.

Como trabajo futuro se propone identificar y desarrollar nuevas funcionalidades para contribuir al monitoreo activo, así como ampliar el rango de recursos a monitorizar por la aplicación.

Agradecimientos

Los autores desean agradecer al colectivo del Centro del Software Libre (CESOL) de la Universidad de las Ciencias Informáticas por los aportes realizados a la presente investigación.

Referencias

Aniel. 2013. <https://www.aniel.es/importancia-de-las-tic-para-la-gestion-empresarial/>. [En línea] 7 de agosto de 2013.

Ana Gardey, Julián Pérez Porto. 2013. Definicion.de. [En línea] 2013. <https://definicion.de/monitoreo/>.

- García Salas, Jorge Stteeven y Roa Piñeros, Camilo Andres. 2020. Diseño de una herramienta de monitoreo y control de servidores utilizando como eje principal cacti. Aplicado a una pyme mediana. Universidad Cooperativa de Colombia, Facultad de Ingeniería, Bogotá DC. Recuperado de https://repository.ucc.edu.co/bitstream/20.500.12494/16571/3/2020-Herramienta_monitoreo_servidores.pdf
- González, Yanette Díaz y Romero, Yenisleidy Fernández. 2012. Patrón Modelo-Vista-Controlador. s.l.: Revista Telem@tica, 2012. vol. 11.
- Ligus, S. (2013). Effective Monitoring and Alerting. (M. Oram, Andy & Hendrickson, Ed.) (2013th ed.). Gravenstein Highway North, Sebastopol. <https://doi.org/10.1016/j.jcp.2015.01.030>
- Nova Servidores. (2016). Recuperado de <https://www.uci.cu/investigacion-y-desarrollo/productos/nova/nova-servidores>
- Pérez Villazón, Yadiel (2018). Solución tecnológica para el monitoreo del sistema de recuperación de información de la Plataforma C.U.B.A. Tesis para optar por el título de Máster en Informática Aplicada, Universidad de la Ciencias Informáticas, La Habana. Recuperado de <https://repositorio.uci.cu/bitstream/123456789/7911/1/documento-v19.0.pdf>
- Pressman, Roger 2011. Ingeniería de Software, un enfoque práctico. s.l.: McGraw-Hill Companies, 2011
- Romero, M.Sc. Gerardo Junco. 2018. Monografias.com. Monografias.com. [En línea] 2018. <https://www.monografias.com/trabajos95/recursos-red-y-su-monitoreo/recursos-red-y-su-monitoreo.shtml>.

Conflicto de interés

El autor autoriza la distribución y uso de su artículo.

Contribuciones de los autores

1. Conceptualización: Hanny Valdés Hernández y David Hiram Hernandez Peña
2. Curación de datos: Hanny Valdés Hernández y David Hiram Hernandez Peña
3. Análisis formal: Hanny Valdés Hernández y David Hiram Hernandez Peña
4. Adquisición de fondos: David Hiram Hernandez Peña
5. Investigación: Hanny Valdés Hernández y David Hiram Hernandez Peña
6. Metodología: Hanny Valdés Hernández y David Hiram Hernandez Peña

7. Administración del proyecto: Hanny Valdés Hernández
8. Recursos: David Hiram Hernandez Peña
9. Software: David Hiram Hernandez Peña y Hanny Valdés Hernández
10. Supervisión: Hanny Valdés Hernández
11. Validación: David Hiram Hernandez Peña
12. Visualización: Hanny Valdés Hernández y David Hiram Hernandez Peña
13. Redacción – borrador original: Hanny Valdés Hernández
14. Redacción – revisión y edición: Hanny Valdés Hernández

^a Programa informático que registra la información que envían los periféricos, así como la actividad realizada en un determinado ordenador.

^b Protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP.

^c Es un estándar que define objetos actuales e históricos de control, permitiendo que usted capture la información en tiempo real a través de la red entera.

^d Herramienta de monitorización de ancho de banda basado en tecnología web.

^e En informática, un script es un guion o conjunto de instrucciones que permiten la automatización de tareas creando pequeñas utilidades.

^f Por sus siglas del inglés Central Processor Unit, al español Unidad Central de Procesamiento, es un componente básico de la computadora

^g El servicio de mensajes cortos o servicio de mensajes simples, más conocido como SMS, es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, conocidos como mensajes de texto, entre teléfonos móviles.

^h Consultar la página: <https://nodejs.org/es/>