



Temática : Tecnologías de software libre

Herramienta para la administración remota de tareas programadas en Nova 7.0

Tool for the remote management of scheduled tasks in Nova 7.0

Sueyaile Toledo Ruiz

Centro de Software Libre, Facultad José Martí, Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, km 2 ½, Torrens, Boyeros, La Habana, CP. 19370. stoledo@uci.cu

* Autor para correspondencia: stoledo@uci.cu

Resumen

En los últimos años el desarrollo de aplicaciones que facilitan el servicio de administración de tareas programadas en Linux ha ido creciendo paulatinamente, ejemplo de ello lo son las herramientas Webmin y Crontab-ui. Con el fin de asegurar la soberanía tecnológica e impulsar la innovación nacional, Cuba ha estado llevando a cabo la migración paulatina hacia Software Libre y las tecnologías de código abierto, proceso en el cual interviene con sus soluciones informáticas el Centro de Software Libre, perteneciente a la Universidad de las Ciencias Informáticas. Nova es una distribución cubana de GNU/Linux y una de estas soluciones, la cual permite realizar las acciones propias de cualquier sistema operativo. No obstante, el servicio de administración de tareas programadas en Nova se realiza de forma manual mediante la introducción de comandos en una interfaz de consola, lo que resulta dificultoso para aquellos usuarios que no tengan dominio suficiente sobre los mismos. La presente investigación tuvo como objetivo implementar una herramienta informática para la administración remota de tareas programadas en Nova 7.0. Para su desarrollo se utilizó como guía en todo el proceso de desarrollo de software la metodología Proceso Unificado Ágil ajustada para la Universidad de las Ciencias Informáticas; así como diferentes tecnologías libres y la arquitectura N-Capas en su variante tres capas. Además, se realizaron pruebas de software para comprobar el correcto funcionamiento de la herramienta, demostrando que la misma cumple con las necesidades establecidas para Nova 7.0 y permite un mejor manejo en la configuración de las tareas programadas.

Palabras clave: tareas programadas, administración remota, Cron, Crontab, Nova 7.0.

Abstract

In recent years the development of applications that facilitate the task management service scheduled in Linux has been growing gradually, an example of which are the Webmin and Crontab-ui tools. In order to ensure technological sovereignty and boost national innovation, Cuba has been carrying out the gradual migration to Free Software and open source technologies, a process in which the Free Software Center, which belongs to the University of Computer

Science. Nova is a Cuban distribution of GNU / Linux and one of these solutions, which allows you to perform the actions of any operating system. However, the task management service programmed in Nova is done manually by entering commands in a console interface, this makes it difficult for those users who do not have sufficient mastery over them. The purpose of this research was to implement a computer tool for the remote management of scheduled tasks in Nova 7.0, for its development the Agile Unified Process variation methodology for the University of Computer Science was used as a guide in the entire software development process; different free technologies and the N-Layers architecture was used in its three-layer variant. In addition, software tests were carried out to check the correct functioning of the tool, demonstrating that it meets the needs established for Nova 7.0 and allows for better management in the configuration of scheduled tasks.

Keywords: *scheduled tasks, remote management, Cron, Crontab, Nova 7.0.*

Introducción

Con el creciente uso de las Tecnologías de la Información y las Comunicaciones (TIC), la información fluye sin fronteras, pues cada día es necesario que grandes volúmenes de datos puedan ser almacenados y gestionados, y Cuba no se encuentra excepto de ello (Rodríguez and Vera, 2015). En la última década en nuestro país se ha estado realizando un arduo proceso de informatización de la sociedad, el cual ha ido encaminado y enfocado a los pilares fundamentales del proceso revolucionario: la salud, la educación y la cultura. Con el fin de asegurar la soberanía tecnológica e impulsar la innovación nacional, se lleva a cabo la migración paulatina hacia Software Libre y las aplicaciones de código abierto, debido a la necesidad que esto representa para la sociedad cubana en los aspectos políticos, económicos, sociales y tecnológicos (Tamayo, 2014).

Con el objetivo de apoyar la informatización del país, en la Universidad de las Ciencias Informáticas (UCI), fue creado el Centro de Software Libre (CESOL), líder en los procesos de migración hacia tecnologías libres y de código abierto en Cuba. El mismo realiza acciones como el desarrollo de aplicaciones de escritorio, herramientas de soporte a dicho proceso y la migración de los servicios telemáticos, los cuales utilizan generalmente técnicas de procesamiento de la información de forma remota. Dentro de dichos servicios telemáticos se encuentra el servicio de planificación de tareas (Alcolea, 2017).

El servicio de planificación de tareas permite realizar, con cierta periodicidad, determinadas acciones o tareas, algunas de ellas internas del Sistema Operativo, otras definidas por el administrador, e incluso algunas definidas por un usuario con los privilegios adecuados. La necesidad de este tipo de herramientas viene dada, en principio, tanto por el funcionamiento interno del sistema operativo como por la necesidad del administrador de garantizar un aceptable funcionamiento del sistema. En el caso de Linux, la planificación se realiza a través de Cron, que no es más que un servicio que administra los procesos que se ejecutan en segundo plano y en intervalos programados por los usuarios, resultando de gran utilidad para los usuarios que interactúan con este, debido a que favorece la gestión de tareas de forma desatendida (Alcolea, 2017).

El desarrollo de herramientas que ofrecen este servicio en Linux se ha ido incrementando progresivamente. Las tareas programadas permiten realizar acciones como copias de seguridad periódicas o controlar el sistema y ejecutar scripts personalizados, entre otras tareas. Existen diversas aplicaciones que implementan este servicio y se encuentran basadas en tecnologías libres, tomando como ejemplo: Webmin (Webmin, 2016), Crontab-ui (Wallen, 2015) y Nova NAS (Libre, 2015). Estas herramientas brindan al usuario la posibilidad de crear, editar y eliminar tareas planificadas, mediante una interfaz gráfica amigable, sin necesidad de tener que acceder a la consola para su configuración; pero a su vez poseen desventajas como: no brindan la posibilidad de administrar remotamente las tareas programadas o no abarcan todos los parámetros del cron.

Actualmente el servicio de administración de tareas programadas en la distribución cubana de GNU/Linux Nova se realiza de forma manual mediante la introducción de comandos a través de una interfaz de consola, la que resulta poco amigable e intuitiva para los usuarios, provocando que el trabajo con la misma resulte complejo. Además, esto trae como consecuencia errores humanos en la configuración de las tareas, un mayor empleo de tiempo para aquellos usuarios que no dominen los comandos y su funcionamiento; de igual manera ocurre con aquellos administradores que tienen que acceder de máquina en máquina cada vez que tenga que configurar las tareas a un conjunto de computadoras.

Con el objetivo de dar solución a la problemática planteada anteriormente la presente investigación propone la implementación de una herramienta informática para administrar de forma remota las tareas programadas en Nova 7.0 y para ello se toma como referencia las características y funcionalidades de las herramientas antes mencionadas.

Materiales y métodos o Metodología computacional



Para el desarrollo de la investigación se utilizó la metodología de desarrollo de software AUP (*Agile Unified Process*, Proceso Unificado Ágil) para la UCI, la cual es una variación de AUP que logra estandarizar el proceso de desarrollo de software en los proyectos productivos de la universidad, además de convertirse en la metodología rectora de su desarrollo productivo, ya que se adapta perfectamente al ciclo de vida de la actividad productiva de los diferentes centros de la institución (Rodríguez, 2015).

En el desarrollo de la investigación se utilizaron los siguientes métodos científicos:

Métodos teóricos:

- Analítico-Sintético: permite realizar un análisis del desarrollo y funcionamiento de las herramientas de administración remota de tareas programadas, para así sintetizar en sus principales aspectos, funcionalidades y características.
- Histórico-Lógico: permite hacer un análisis completo de la evolución de las diferentes herramientas de administración de tareas programadas existentes, seleccionando las más utilizadas, para así obtener las características e información necesaria para el diseño de la propuesta de solución.

Métodos empíricos:

- Observación: se realiza con el objetivo de constatar el estado actual de la problemática mediante la observación al proceso de seguimiento de la administración de tareas programadas en entornos libres.
- Entrevista: se realizan entrevistas al cliente y a especialistas del Centro CESOL para obtener la información necesaria referente a los requisitos de la herramienta a desarrollar.
- Encuesta: se realiza con el objetivo de recopilar información en cuanto al nivel de satisfacción de la propuesta de solución.

Tecnologías para la implementación

Para el proceso de desarrollo de la herramienta de administración remota de tareas programadas para Nova 7.0 se han seleccionado un conjunto de herramientas y tecnologías que ayudan al diseño, la modelación e implementación de la solución propuesta, con el objetivo de obtener resultados satisfactorios. Entre los lenguajes utilizados en la implementación de la propuesta de solución se encuentran HTML5, CSS3 y JavaScript. Como *frameworks* o marcos de trabajos se emplea Electron v8.0.0, Vue.js v4.2.2 y Vuetify como marco de diseño. El Entorno Integrado de Desarrollo (IDE), seleccionado para la construcción de la herramienta es Visual Studio Code v1.42.1 y Git como herramienta para el control de versiones.

Arquitectura de la herramienta



Se define como arquitectura de la propuesta de solución la arquitectura N-capas, en su variante 3 capas, ya que brinda una organización jerárquica donde cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior. Las capas son las siguientes: Presentación, Lógica de Negocio y Acceso a Datos. La primera es la capa encargada de la representación visual de la herramienta, informar sobre la situación de los procesos de negocio e implementación de las reglas de validación de interfaz. Las interfaces de usuario se comunican con la capa de Lógica de Negocio. La segunda capa es donde se alojan mayoritariamente las clases e interactúa con la capa Presentación para recibir y procesar las peticiones del usuario y enviar las respuestas a la capa de Acceso a Datos. La capa de Acceso a Datos tiene asignada la responsabilidad de contener el código necesario para el manejo de datos.

Pruebas de software

Se realizaron un conjunto de pruebas de software para verificar el comportamiento de la herramienta entre ellas las pruebas unitarias, funcionales y de aceptación.

Las pruebas unitarias se realizaron mediante la aplicación del método caja blanca y la técnica del camino básico. Las pruebas unitarias, en las que se basa este trabajo, están específicamente encaminadas al enfoque de caja blanca ya que centran el proceso de verificación en la menor unidad del diseño: el módulo, la clase, el método, entre otros. Estas utilizan la descripción del diseño detallado como guía, para probar los caminos de control importantes, con el fin de descubrir errores en la unidad (Rojas, Pérez, Delgado and 2019). Luego de realizada las pruebas unitarias se obtuvo como resultado que el código generado está libre de códigos innecesarios y ciclos infinitos. Además, se comprobó que todas las sentencias de código se ejecutan al menos una vez, por lo que el flujo de trabajo de las funcionalidades para la herramienta de administración remota de tareas programadas en Nova 7.0 es correcto.

Las pruebas funcionales son aquellas que se llevan a cabo sobre la interfaz del software sin prestar atención al código, por lo que los casos de prueba son creados con el objetivo de demostrar que la entrada es aceptada de forma adecuada y que se produce una salida correcta. El diseño de esta prueba se realiza con la intención de detectar funciones incorrectas o ausentes, errores en accesos a bases de datos externas, errores de interfaz, errores de rendimiento, y errores de inicialización y de terminación (Barrientos, 2019). Las pruebas funcionales se realizaron mediante el método de caja negra y partición de equivalencia. Para dicha técnica se definen las variables válidas y no válidas para cada entrada del sistema. Las pruebas funcionales se realizaron en 2 iteraciones, en la primera se identificaron 17 no conformidades, de ellas 12 de errores ortográficos y 5 de validación. Durante la segunda iteración no se identificaron no conformidades.

Las pruebas de aceptación son especificadas por el cliente y se centran en las características y funcionalidades generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las Historias de usuario que se han implementado como parte de la liberación del software. El objetivo de esta prueba es encontrar los errores y las omisiones en la definición de los requisitos del sistema, debido a que los datos reales ejercitan el sistema en diferentes formas a partir de datos de prueba (Aquia, 2019). Como resultado de la aplicación de las pruebas de aceptación se obtuvo una evaluación satisfactoria por parte del cliente. Lo que evidencia el cumplimiento de los objetivos propuesto para la realización de esta prueba.

Resultados y discusión

La herramienta desarrollada posee una totalidad de 15 requisitos, de ellos 9 son requisitos funcionales (RF) y 6 requisitos no funcionales (RNF). La Tabla 1 muestra el listado de los requisitos funcionales de la propuesta de solución. La complejidad de los requisitos funcionales se obtuvo mediante el producto de trabajo “Evaluación de requisitos”, propuesto en el expediente de proyecto 5.0 para la actividad productiva de la UCI.

Tabla 1. Listado de requisitos funcionales.

Número	Requisito	Descripción	Complejidad	Prioridad
RF1	Conectarse remotamente	La herramienta permite conectarse remotamente a otra máquina.	Alta	Alta
RF2	Iniciar servicio <i>cron</i>	La herramienta permite iniciar el servicio <i>cron</i> , el cual se encuentra instalado automáticamente en el sistema.	Media	Alta
RF3	Reiniciar servicio <i>cron</i>	La herramienta permite reiniciar el servicio <i>cron</i> .	Media	Media
RF4	Detener servicio <i>cron</i>	La herramienta permite detener el servicio <i>cron</i> .	Media	Media
RF5	Crear tarea programada	La herramienta permite crear una tarea programada.	Alta	Alta
RF6	Mostrar tarea programada	La herramienta muestra una tarea programada con sus campos (nombre, minutos, horas, días de la semana, día del mes, mes, usuario y comando).	Media	Media

RF7	Listar tareas programadas	La herramienta muestra un listado con todas las tareas programadas que se encuentran creadas hasta el momento.	Media	Media
RF8	Editar tarea programada	La herramienta permite editar una tarea programada que ha sido seleccionada previamente.	Alta	Alta
RF9	Eliminar tarea programada	La herramienta permite eliminar una tarea programada que ha sido seleccionada previamente.	Media	Media

La definición de los requisitos no funcionales de la propuesta de solución se realizó mediante el estándar de calidad ISO-25010 (*International Organization for Standardization*, Organización Internacional de Normalización), propuesto en el producto de trabajo “Especificación de requisitos de software” del expediente de proyecto utilizado en la actividad productiva de la UCI. A continuación, se presenta en la Tabla 2 el listado de los requisitos no funcionales de la propuesta de solución.

Tabla 2. Listado de requisitos no funcionales.

Número	Requisito	Descripción
RNF1	Requisito de compatibilidad	La herramienta para administrar remotamente las tareas programadas debe ser compatible con Nova 7.0.
RNF2	Requisito de seguridad	A la herramienta solo tendrán acceso los usuarios administradores.
RNF3	Requisito de mantenibilidad	La herramienta debe hacer uso de los estándares de codificación definidos para la herramienta.
RNF4	Requisito de usabilidad 1	En la herramienta se deben visualizar todos los mensajes en idioma español. La tipografía debe ser uniforme, de un tamaño adecuado.
RNF5	Requisito de usabilidad 2	La herramienta debe ofrecer una interfaz amigable, fácil de operar. Las interfaces deben poseer un diseño sencillo, con pocas entradas, permitiendo un balance adecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización de la misma.
RNF6	Restricciones del diseño e implementación	La herramienta se debe desarrollar utilizando el lenguaje del lado del cliente: HTML5, CSS3 y JavaScript., así como los marcos de trabajo <i>Vuejs</i> v 4.2.2 y <i>Electron</i> v.8.0, como marco de diseño <i>Vuetify</i> y como entorno de desarrollo Visual Studio Code v 1.42.

A continuación, se muestra una comparación entre las aplicaciones informáticas para la administración de tareas programadas mencionadas en el comienzo de la investigación. Para ello se definieron una serie de indicadores, de acuerdo con los requisitos necesarios para el desarrollo de la propuesta de solución y el contexto del negocio, además de tenerse en cuenta las necesidades del cliente, ver Tabla 3.

Tabla 3. Definición de criterios y su clasificación.

Clasificación	Sí cumple	No cumple
Criterios para el análisis		
Basada en software libre	Está basada en software libre	No está basada en software libre
Integración con Nova 7.0	Permite la integración con Nova 7.0	No permite la integración con Nova 7.0
Administración remota por protocolo SSH	Permite la administración de forma remota	No permite la administración de forma remota
Totalidad de parámetros del <i>cron</i>	Abarca todos los parámetros del <i>cron</i>	No abarca todos los parámetros del <i>cron</i>
Detección de cambios manuales en los ficheros de configuración del <i>cron</i>	Detecta todos los cambios realizados en los ficheros del <i>cron</i>	No detecta todos los cambios realizados en los ficheros del <i>cron</i>

Con la finalidad de demostrar el cumplimiento de los requerimientos de la propuesta de solución, en la Tabla 4 se muestra la comparación entre las aplicaciones informáticas para la administración de tareas programadas estudiadas al comienzo de la investigación y la propuesta de solución.

Tabla 4. Análisis comparativo entre aplicaciones de administración remota de tareas programadas.

Criterios para el análisis	Sistemas informáticos para la administración de tareas programadas			
	Crontab-ui	Webmin	Nova NAS	Propuesta de solución
Basada en software libre	Sí	Sí	Sí	Sí
Integración con Nova 7.0	No	Sí	Sí	Sí
Administración remota por protocolo SSH	Sí	Sí	Sí	Sí
Totalidad de parámetros del <i>cron</i>	Sí	Sí	No	Sí
Detección de cambios manuales en los ficheros de configuración del <i>cron</i>	Sí	No	Sí	Sí



Al comparar las aplicaciones estudiadas con la herramienta informática desarrollada en la presente investigación para administrar de forma remota las tareas programadas, se arroja como resultado que esta es superior en cuanto a los criterios tenidos en cuenta. De esta forma se denota el aporte de la propuesta de solución para cubrir necesidades de Nova 7.0.

Conclusiones

Con la realización de la presente investigación se brinda solución a los objetivos trazados, obteniéndose como principales resultados: el análisis del marco teórico referencial sobre la administración de tareas programadas en GNU/Linux y el estudio de aplicaciones informáticas que permiten este servicio demostraron la necesidad de implementar una herramienta que permita la administración remota de tareas programadas en Nova 7.0. La aplicación de las pruebas permite corregir las no conformidades detectadas, validando así todas las funcionalidades concebidas por el cliente y logrando una herramienta de escritorio con calidad y seguridad. Como resultado de la implementación se obtuvo una herramienta que le permite realizar al usuario una mejor configuración de las tareas programadas en Nova 7.0 sin necesidad de acceder a la introducción de líneas de comando mediante la consola.

La herramienta desarrollada le permitirá a los administradores o usuarios con ciertos privilegios realizar con mínimos esfuerzos la configuración de las tareas programadas en Nova 7.0 mediante una interfaz amigable e intuitiva. Además, brinda la posibilidad de informatizar algunos procesos del sistema de forma automática y puede ser utilizada por empresas o proyectos que no posean grandes recursos de hardware pues solo se necesita una computadora con distribución cubana de GNU/Linux Nova 7.0, SSH y Electron instalados.

A partir de la experiencia obtenida en el desarrollo de la investigación, la autora recomienda: utilizar el contenido de la investigación como base de referencia para el desarrollo de futuras herramientas para administrar tareas programadas en otras versiones Nova y en otras distribuciones de GNU/Linux. Así como crear nuevas funcionalidades que le brinde a la herramienta la tolerancia a fallos y la recuperación ante errores.



Referencias

- Alcolea, W. E. (2017). *Módulo de planificación de tareas para la Herramienta de Migración y Administración de Servicios Telemáticos*. Universidad de las Ciencias Informáticas, La Habana.
- Alejandro Rodríguez Toledo, O. V. (2015). *Módulo para la ejecución automática scripts en los clientes desde el servidor de GRHS*. Universidad de las Ciencias Informáticas, La Habana.
- Aquia, S. A. (2019). *Módulo para el cifrado de los datos informáticos almacenados en NovaNAS*. Universidad de las Ciencias Informáticas, La Habana.
- Barrientos, M. d. (2019). *Módulo para la implementación de mecanismos de bloqueo a las conexiones remotas que intentan accesos por fuerza bruta para Nova-LTSP*. Universidad de las Ciencias Informáticas, La Habana.
- Dania Mailen Rojas-Robert, Z. P.-M.-D. (1 de Agosto de 2019). Generador de valores interesantes para casos de pruebas unitarias. *Ingeniería Industrial*, 40(2). Recuperado el 2 de Agosto de 2021
- Libre, C. d. (2015). *Administración de un servidor de almacenamiento conectado a la red con Nova*. Universidad de las Ciencias Informáticas, La Habana.
- Rodríguez, T. (2015). *Metodología de desarrollo para la actividad productiva de la UCI*. Universidad de las Ciencias Informáticas, La Habana.
- Tamayo, Y. M. (2014). *Administración del servicio antivirus desde la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST)*. Universidad de las Ciencias Informáticas, La Habana.
- Wallen, J. T. (Abril de 2015). Recuperado el 2 de Diciembre de 2019, de <https://www.techrepublic.com/article/how-to-install-crontab-ui-for-remote-use/>
- Webmin. (2016). *Webmin*. Recuperado el 2 de Diciembre de 2019, de <http://www.webmin.com/intro.html>.