



Temática: Software Libre

## Gestión de copias de seguridad en Nova 360 *Managing backups in Nova 360*

Miosotis Toledo Rodriguez

<sup>1</sup> Centro de Software Libre, Universidad de las Ciencias Informáticas. Carretera de San Antonio de los Baños, km 2 ½,

Torrens, Boyeros, La Habana, Cuba. [mtoledo@uci.cu](mailto:mtoledo@uci.cu)

---

### Resumen

La información digital está en constante crecimiento y tal aumento exige grandes capacidades de almacenamiento. La computación en la nube, a través de Internet, ha facilitado el consumo de estos volúmenes de información mediante el uso de aplicaciones como OwnCloud y NextCloud. Cuba en los últimos años ha avanzado considerablemente en el uso de las Tecnologías de la Información y las Comunicaciones como parte de la estrategia de informatización de la sociedad. Con el objetivo de lograr la soberanía tecnológica, se desarrolla un proceso de migración hacia software libre en el que se emplean soluciones informáticas desarrolladas por el Centro de Software Libre, perteneciente a la Universidad de las Ciencias Informáticas. Nova 360 es una de estas soluciones; tiene como objetivos ofrecer servicios en la nube, permitir la sincronización con múltiples dispositivos de datos y compartir archivos de forma pública. Sin embargo, para realizar la gestión de sus copias de seguridad es necesario el uso de comandos en la terminal, lo cual representa una dificultad para aquellos usuarios que no tienen conocimientos avanzados de los comandos de Linux. La presente investigación tuvo como objetivo desarrollar un módulo que gestione las copias de seguridad de Nova 360, para ello se utilizaron tecnologías en su mayoría libres, diferentes patrones de diseño y la arquitectura Modelo-Vista-



Controlador. Con las pruebas aplicadas se comprobó el correcto funcionamiento de Salvias, demostrando que el mismo satisface las necesidades del cliente.

**Palabras clave:** copias de seguridad, información digital, Nova 360, soberanía tecnológica, sistema de gestión

## Abstract

Digital information is constantly growing and such growth requires large storage capacities. Cloud computing, through the Internet, has facilitated the consumption of these volumes of information through the use of applications such as OwnCloud and NextCloud. Cuba in recent years has made considerable progress in the use of Information and Communication Technologies as part of the computerization strategy of society. In order to achieve technological sovereignty, a migration process towards free software is developed in which computer solutions developed by the Free Software Center, belonging to the University of Computer Sciences, are used. Nova 360 is one of these solutions; aims to offer cloud services, allow data synchronization with multiple devices, and public file sharing. However, to manage your backups it is necessary to use commands in the terminal, which represents a difficulty for those users who do not have advanced knowledge of Linux commands. The objective of this research was to develop a module that manages the backup copies of Nova 360, for this purpose, mostly free technologies, different design patterns and the Model-View-Controller architecture were used. With the applied tests, the correct operation of Salvias was verified, showing that it satisfies the client's needs.

**Key words:** backups, digital information, Nova 360, technological sovereignty, management system

---

## Introducción

La información digital está en constante crecimiento y tal aumento exige disponer de grandes volúmenes de almacenamiento para protegerla, ya que es uno de los bienes más importantes con que cuentan los seres humanos para



el desarrollo del conocimiento. La evolución de Internet ha venido acompañada de la posibilidad de consumir estos altos volúmenes de información situados en servidores ubicados en diversos lugares del planeta (MORRISON, 2016). Una de las tecnologías que ha contribuido con este propósito es la computación en la nube: un modelo de prestación de servicios de negocios y tecnologías a través de una red informática que divide en tres categorías, el Software como Servicio (*SaaS, Software as a Service*), Plataforma como Servicio (*PaaS, Platform as a Service*), e Infraestructura como Servicio (*IaaS, Infrastructure as a Service* (TWEEDIE, 2017).

El desarrollo de aplicaciones que ofrecen estos servicios ha tenido un gran auge en la comunidad de software libre, tomando como ejemplo: OwnCloud, OpenStack, NextCloud; las cuales representan una alternativa a aplicaciones propietarias como: Dropbox, iCloud, OneDrive o Google Drive. Estas herramientas ofrecen servicios de nubes personalizadas garantizando la seguridad de los datos almacenados. Permiten hacer copias de seguridad de su sistema de configuración y almacenamiento, las cuales pueden ser restauradas ante un fallo que comprometa la estabilidad del sistema y la pérdida total o parcial de su información (MAURYA, 2018).

Cuba en los últimos años ha avanzado considerablemente en el uso de las Tecnologías de la Información y las Comunicaciones (TIC) para la gestión de los procesos de los diferentes Organismos de la Administración Central del Estado (OACE) y como política de la estrategia para la informatización de la sociedad cubana. Con el objetivo de garantizar la soberanía tecnológica, en estas entidades se desarrolla una migración hacia software libre; en ella se utilizan soluciones informáticas desarrolladas por el Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI). Entre estas soluciones se encuentran: la distribución cubana GNU/Linux Nova (como sistema operativo), Nova Servidores (para brindar servicios telemáticos) y Nova 360 (ofreciendo servicios en la nube).

Nova 360 es una personalización de Owncloud para las instituciones cubanas y posibilita la sincronización con múltiples dispositivos de datos, calendarios, tareas, archivos multimedia, galería de imágenes y otros. Permite compartir archivos entre usuarios o de forma pública. Su desarrollo está centrado en brindar una solución que integre diferentes tecnologías de Nova tales como: NovaDroid (personalización cubana para móviles con Android) y Nova Escritorio.



En la actualidad en Nova 360 para realizar la gestión de copias de seguridad es necesario hacer uso de comandos en la consola del servidor que lo contiene, lo cual representa una dificultad para aquellos usuarios que no dispongan de los conocimientos necesarios en cuanto al uso de los comandos precisos, su sintaxis y funcionamiento. Asimismo, al tener menor experiencia deben emplear mayor tiempo en la escritura de las instrucciones para efectuar las copias de seguridad y la restauración de estas. El empleo de la consola también genera cierta resistencia en los usuarios ya que resulta poco amigable e intuitiva.

La situación descrita anteriormente provoca que este proceso no se pueda realizar con la sistematicidad requerida, ya que es una tarea engorrosa. Además, ante la ocurrencia de un fallo que comprometa la persistencia de los datos, ya no se podría contar con la información almacenada, por lo que para guiar la investigación se definió como objetivo: Desarrollar un módulo para realizar la gestión de las copias de seguridad en Nova 360 que agilice el proceso de planificación de respaldo y restauración de la información que se tiene almacenada.

## **Materiales y métodos o Metodología computacional**

Con el objetivo de conocer cómo se efectúa el proceso de gestión de copias de seguridad se analizaron varios sistemas dedicados a la gestión de copias de seguridad: Bacula rsync, fwbackups y Box Backup. El análisis realizado evidenció que esas aplicaciones no cumplían en su totalidad las necesidades de Nova 360 para realizar la gestión de copias de seguridad, demostrando la importancia de desarrollar un módulo que aproveche las mejores funcionales de estas aplicaciones. Una vez realizado este estudio se procedió a determinar cuál sería la metodología de desarrollo de software más idónea para poder darle solución al problema planteado. Siguiendo la política de la Universidad de las Ciencias Informáticas se utilizó la metodología de desarrollo de software Variación de AUP para la UCI, ya que esta metodología logra estandarizar el proceso productivo en los proyectos de la universidad.

Esta metodología cuenta con 3 fases: Inicio, Ejecución y Cierre (RODRÍGUEZ, 2015). La presente investigación se desarrolla durante la fase de Ejecución, pues la misma propicia que se ejecuten las actividades necesarias para desarrollar el software. Se transita por las disciplinas: Requisitos (teniendo en cuenta las características del proyecto se



empleó el cuarto escenario de la metodología que sugiere realizar Historias de usuarios para describir esta disciplina), Análisis y Diseño, Pruebas internas y Pruebas de aceptación.

Para modelar el diseño arquitectónico se empleó el patrón arquitectónico Modelo-Vista-Controlador, quedando representado en el modelo las clases entidades y de acceso a datos, en el controlador las clases controladoras de la solución. En las vistas, las plantillas asociadas a cada una de las vistas de la solución. Además, se utilizan los patrones de diseño GRASP (bajo acoplamiento, alta cohesión, experto y creador), e Inyección de dependencias.

Para el modelado de los diagramas de clase, de despliegue y modelo de datos se utilizó la herramienta CASE Visual Paradigm v8.0 haciendo uso del lenguaje UML v5.0. durante el desarrollo del módulo Salvas se empleó PhpStorm v2018.2.5 (Jetbrains, 2018) y como sistema gestor de base de datos MySQL v5.7.11 (Oracle Corporation, 2018). Para el control de las versiones se tuvo en cuenta la herramienta Git (CHACON & STRAUB, 2014), y como servidor de aplicaciones web Apache v2.4 (Apache, 2018). Mientras que fueron usados los lenguajes Php v7.0.8 (GONZÁLES, 2015), JavaScript (Mozilla, 2018), HTML5 (PÉREZ & GARDEY, 2018) y CSS3 (W3C, 2018).

Durante la implementación se tuvieron en cuenta las reglas de oro que Roger Pressman define para el diseño e implementación de interfaces gráficas de usuario (PRESSMAN, 2010). Para comprobar el correcto funcionamiento del módulo Salvas se realizaron pruebas unitarias y funcionales. En las pruebas unitarias se utilizó el método de caja blanca a través de la técnica del camino básico, así como el método de caja negra con la técnica de partición equivalente durante las pruebas funcionales. La aplicación de las pruebas unitarias permitió corroborar que el flujo de trabajo de las funcionalidades implementadas es el correcto, pues cada sentencia se ejecuta al menos una vez. La ejecución de las pruebas funcionales posibilitó identificar las deficiencias de la propuesta de solución y garantizar el cumplimiento de los requisitos definidos.

## Resultados y discusión

Siguiendo el patrón arquitectónico Modelo-Vista-Controlador, en la capa de las vistas se encuentran todas las plantillas asociadas a cada una de las interfaces de la solución. A través de estas interfaces se muestra o se capta la información necesaria utilizando formularios, mensajes de confirmación o de error.

El modelo de datos elaborado para el módulo de gestión de copias de seguridad cuenta con dos tablas que representan la información de las entidades *backup\_salva* y *backup\_server*. En la primera se almacenan todos los datos referentes a la programación de una copia de seguridad, y en la segunda se recogen los datos asociados a los servidores donde se almacenan las copias de seguridad realizadas.

Para describir los requisitos se emplearon historias de usuarios. A continuación, se presenta la correspondiente al RF1. Adicionar planificación de copia de seguridad.

Tabla 1. Descripción de la historia de usuario del RF.1 adicionar planificación de copia de seguridad

HISTORIAS DE USUARIOS	
<b>Número:</b> RF1	<b>Nombre del requisito:</b> Adicionar planificación de copia de seguridad.
<b>Programador:</b> Miosotis Toledo Rodríguez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 25 horas
<b>Riesgo en Desarrollo:</b> No aplica	<b>Tiempo Real:</b> 25 horas
<b>Descripción:</b> El módulo permite al usuario crear la planificación de una copia de seguridad. El usuario selecciona la opción de “Adicionar planificación” del menú superior, luego se activan los criterios de selección de tiempo (minuto, hora, día de la semana, día del mes, mes), un campo para poner el nombre de la planificación y seleccionar la ubicación donde se realizará la copia de seguridad, la cual puede ser remota o local. En caso de que no desee seleccionar ninguna de las ubicaciones anteriores contará con un botón que le envía a la vista en la que crea una ubicación nueva. Una vez llenados los campos se selecciona la opción de “Crear” en el menú inferior. En caso de que el usuario se arrepienta en la creación de una planificación se cuenta con la opción “Cancelar” también en el menú inferior.	
<b>Observaciones:</b>	

## Prototipo elemental de interfaz gráfica de usuario:

El prototipo muestra una interfaz con los siguientes elementos:

- Adicionar planificación:** Botón principal en la parte superior izquierda.
- Planificación 1 / Planificación 2:** Lista de planificación existente.
- Generales:** Campos para Título, Ubicación 1, 2 y 3, con un botón de añadir (+).
- Meses:** Opciones de frecuencia (Cada mes, Meses pares, Meses impares) y un selector de meses (Enero a Mayo).
- Días de la semana:** Opciones de frecuencia (Cada día de la semana, Lunes-Viernes, Sábado y Domingo) y un selector de días (Lunes a Viernes).
- Días del mes:** Opciones de frecuencia (Cada día, Días pares, Días impares) y un selector de días (1 a 5).
- Minutos:** Opciones de frecuencia (Cada minuto, Minutos pares, Minutos impares) y un selector de minutos (0 a 4).
- Horas:** Opciones de frecuencia (Cada hora, Horas pares, Horas impares) y un selector de horas (12 am a 4 am).
- Botones de acción:** Opciones, Crear y Cancelar.

Figura 1. Prototipo de interfaz de usuario correspondiente al RF1. Adicionar planificación de copia de seguridad

Luego de realizar la especificación de los requisitos se efectuó la modelación de los diagramas de clases del diseño por agrupación de requisitos utilizando estereotipos webs. En estos diagramas se evidenció el uso de los patrones de diseño GRASP e Inyección de dependencias. En la siguiente figura se presenta el diagrama de clases del diseño correspondiente al RF1. Adicionar planificación de copia de seguridad.

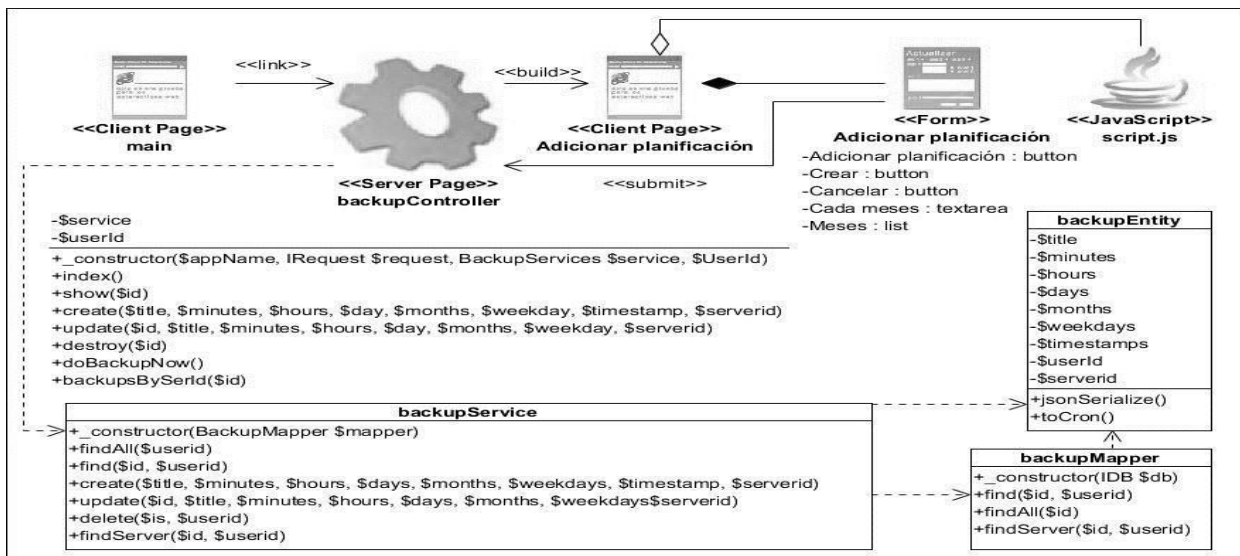


Figura 2. Diagrama de clases del diseño del RF1. Adicionar planificación de copias de seguridad

Siguiendo las exigencias del diseño, así como los requisitos funcionales y no funcionales se desarrolló un sistema que corresponde con lo descrito anteriormente. La siguiente figura es un ejemplo de interfaz de usuario que se hace corresponder con el RF1. Adicionar planificación de copias de seguridad.





Figura 3. Interfaz gráfica de usuario de la planificación de copias de seguridad

Durante la validación de la solución desarrollada se realizaron pruebas unitarias, funcionales y de integración. En las pruebas unitarias se aplicó el método de caja blanca con la técnica de camino básico. Para esta técnica se debe hacer un análisis de la complejidad ciclomática de cada uno de los procedimientos que componen el sistema (PRESSMAN, 2010). Como resultado de la aplicación de esta prueba se obtuvo que el flujo de trabajo de las funcionalidades del módulo para la gestión de copias de seguridad es correcto pues se comprobó que cada sentencia se ejecuta al menos una vez.

Para las pruebas funcionales se utilizó el método de caja negra con la técnica de partición equivalente. En esta técnica se tuvieron en cuenta varios casos de prueba en los que se definieron las clases de equivalencia válidas y no válidas para cada campo de entrada del sistema. La siguiente figura muestra un gráfico con los resultados:

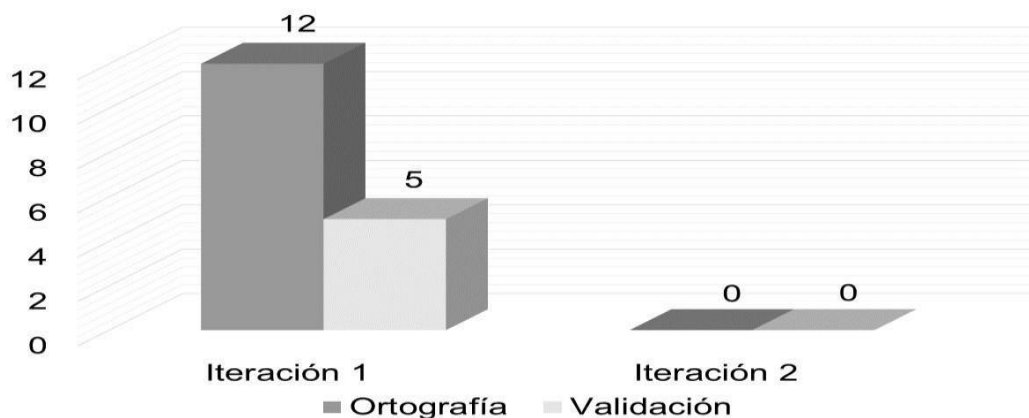


Figura 4. Resultados de las pruebas funcionales

Las pruebas funcionales se hicieron en dos iteraciones, en la primera se identificaron 17 no conformidades. De ellas 12 correspondían a errores ortográficos y 5 a errores de validación. Mientras que en la segunda iteración no se encontraron no conformidades.

Para las pruebas de integración se realizó un diseño de caso de prueba con 3 escenarios que corresponden a las posibles situaciones que se pueden presentar cuando se procede a instalar Salvias en Nova 360. Como resultado se obtuvo que el módulo para la gestión de copias de seguridad se integró de manera correcta a Nova 360, cumpliendo con los requerimientos definidos

## Conclusiones

La presente investigación documentó el proceso de desarrollo del módulo para la gestión de copias de seguridad en Nova 360. Una vez finalizado el mismo se pueden arribar a las siguientes conclusiones:

- ✓ El análisis del marco teórico sobre el proceso de gestión de copias de seguridad para Nova 360, así como el estudio de aplicaciones informáticas para la gestión de copias de seguridad, demostraron la necesidad de desarrollar una aplicación que contribuya con la gestión de copias de seguridad en Nova 360.



- ✓ El análisis y diseño propició la elaboración de un módulo para la gestión de copias de seguridad en Nova 360 permitiendo cumplir con las exigencias del cliente y las restricciones del diseño elaborado. El uso de los patrones de diseño GRASP e Inyección de dependencias y el patrón arquitectónico MVC brindó una mayor calidad al software desarrollado.
- ✓ La aplicación de las pruebas de software permitió evaluar la solución desarrollada y garantizar el correcto funcionamiento del módulo implementado.

## Referencias

- Apache. (30 de abril de 2018). *The Apache Software Foundation*. Recuperado el 31 de octubre de 2018, de <https://www.apache.org/>
- CHACON, S., & STRAUB, B. (2014). *Pro Git* (segunda edición ed.). Apress. Recuperado el 23 de Octubre de 2018
- GONZÁLES, J. (2015). *Desarrollo de sitios web con PHP y MySQL*.
- JetBrains. (25 de Octubre de 2018). *PhpStorm*. Recuperado el 25 de diciembre de 2018, de <https://www.jetbrains.com/phpstorm/>
- MAURYA, R. (19 de Febrero de 2018). *H2S MEDIA*. Recuperado el 24 de octubre de 2018, de 6 Best Open source personal Cloud software to setup cloud storage: <https://www.how2shout.com/tools/best-opensource-personalcloud-software-cloud-storage.html>
- Mozilla. (Octubre de 2018). *Moz://a*. Recuperado el 27 de noviembre de 2018, de MDN web docs: [https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca\\_de\\_JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript)
- Oracle Corporation. (2018). *Oracle*. Recuperado el 23 de diciembre de 2018, de <http://www.oracle.com/us/products/mysql/overview/index.html>
- PÉREZ, J., & GARDEY, A. (Octubre de 2018). *Definiciones.de*. Recuperado el 26 de enero de 2019, de Definicion de HTML: <https://definicion.de/html/>
- PRESSMAN, R. (2010). *Ingeniería de software. Un enfoque práctico* (séptima edición ed.). México: McGraw-Hill.
- RODRÍGUEZ, T. (2015). *Metodología de desarrollo para la actividad productiva de la UCI*. La Habana.



- TWEEDIE, M. (5 de Diciembre de 2017). *WorkingMouse*. Recuperado el 21 de octubre de 2018, de Software Licensing and Cloud Service Models: <https://workingmouse.com.au/innovation/software-licensing-andcloudservice-models>
- W3C. (Diciembre de 2018). *W3C*. Recuperado el 23 de enero de 2019, de Cascading Style Sheets: <https://www.w3.org/Style/CSS/>