



**Universidad de las Ciencias Informáticas**

**Facultad 1**

**Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas**

Título

Módulo para la administración de antivirus en Nova NAS

Autor

Lázaro Miguel de las Cagigas Barrios

Tutores

Ing. Yasiel Pérez Villazón

Ing. Nurisel Palma Pérez

**La Habana, Cuba, 2016**

**“Año 58 de la Revolución”**



LO QUE SABEMOS ES UNA GOTTA DE AGUA; LO QUE IGNORAMOS ES EL OCÉANO.

LOS HOMBRES CONSTRUIMOS DEMASIADOS MUROS Y NO SUFICIENTES PUENTES.

SI HE HECHO DESCUBRIMIENTOS INVALUABLES HA SIDO MÁS POR TENER PACIENCIA QUE CUALQUIER OTRO TALENTO.

SI CONSIGO VER MÁS LEJOS ES PORQUE HE CONSEGUIDO AUPARME A HOMBROS DE GIGANTES.

ISAAC NEWTON

## Dedicatoria

*A mi madre Aurora y mi padre Lázaro Miguel por ser las personas que han estado a mi lado en todo momento brindándome su amor y apoyo.*

*A mis hermanas, en especial a Yudelys y Yanidelys.*

## Agradecimientos

*A mi madre por darme tanto amor, por ser la luz de mis días y el motivo para seguir adelante.*

*A mi padre por su amor y confianza.*

*A mi familia y los que han sido como si lo fueran, en especial a mi abuela Juana, Camilo y mi padrino Pedrito.*

*A mis hermanas, en especial a Yanidelys y Yudelys.*

*A mi novia Ana Bella que siempre ha estado cuando más la necesito.*

*A Gedric por ser como un hermano y estar en todo momento.*

*A los profesores que formaron parte de mi formación, en especial a mis tutores sin los cuales no hubiese podido llevar a cabo este trabajo.*

*A los hermanos que te da la vida en especial a Yainel, Julio, Ernesto y Yulian.*

*A todos mis compañeros de grupo que han sido cómplices de esta trayectoria, en especial a Felipe, Yordy, Yosmay, Yosel, José y Agnes.*

*A todo aquel que de una forma u otra formó parte de mi formación y camino.*

## Declaración jurada de autoría

Declaro por este medio que yo Lázaro Miguel de las Cagigas Barrios, con carné de identidad 92100533143, ser el único autor de este trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2015.

---

Firma del autor

---

Firma del tutor

---

Firma del tutor

## Resumen

Como parte del proceso de desarrollo de software y la migración nacional a software libre surge Nova NAS, una aplicación web desarrollada en el Centro de Software Libre de la Universidad de las Ciencias Informáticas para la administración de un servidor de almacenamiento en la red. Esta cuenta con ClamAV para la protección contra el accionar de programas malignos; pero ClamAV no es un producto nacional y su uso no está autorizado en el país, según plantea la Resolución No. 127/2007 del Ministerio de Información y las Comunicaciones. El presente trabajo de diploma tiene el objetivo de desarrollar un módulo para Nova NAS que permita la administración de un antivirus que cumpla con la Resolución No. 127/2007 del MIC. Para ello se analizan los servicios antivirus que cumplen con dicha resolución y pudiesen ser administrados desde Nova NAS, seleccionando el antivirus SAV Unix como el software más indicado. Además, se documentan las tecnologías, herramientas, lenguajes a utilizar en la construcción del módulo y la definición de los elementos necesarios para el exitoso desarrollo del mismo, así como los artefactos requeridos por la metodología de desarrollo SXP. El principal resultado de la investigación consiste en la implementación de un módulo que permite administrar el servicio antivirus SAV Unix, sus configuraciones generales (agregar licencia y configurar la actualización) y permitir el escaneo de directorios, ofreciendo la posibilidad de observar los resultados del mismo.

**Palabras clave:** administración, antivirus, SAV Unix, Nova NAS.

# Índice

Introducción .....	1
Capítulo 1. Fundamentación teórica del módulo antivirus para Nova NAS.....	5
1.1 Conceptos Fundamentales .....	5
1.1.1 Virus Informáticos .....	5
1.1.2 Antivirus.....	6
1.2 Análisis de la Resolución No. 127/2007 del MIC.....	7
1.3 Antivirus de producción nacional u otros autorizados oficialmente para su uso en el país.....	8
1.3.1 Análisis de la entrevista .....	8
1.3.2 Antivirus de producción nacional .....	8
1.3.3 Antivirus internacionales oficialmente autorizados .....	9
1.4 Selección del antivirus.....	12
1.5 Descripción de Nova NAS .....	13
1.5.1 Conceptos asociados .....	13
1.5.2 Arquitectura .....	13
1.5.3 Requisitos o consideraciones para integrarle un módulo.....	14
1.6 Administración de antivirus.....	15
1.6.1 Administración del servicio antivirus SAV Unix .....	15
1.7 Herramientas y tecnologías a emplear .....	17
1.7.1 Lenguajes de programación utilizados para el desarrollo del módulo .....	17
1.7.2 Framework de desarrollo.....	18
1.7.3 Entorno de Desarrollo Integrado (IDE) .....	18
1.7.4 Herramienta CASE para el modelado.....	19



1.8 Metodología de desarrollo utilizada .....	19
1.9 Conclusiones parciales .....	20
Capítulo 2. Análisis y diseño del módulo para Nova NAS.....	22
2.1 Propuesta del módulo a desarrollar.....	22
2.2 Características y cualidades.....	22
2.2.1 Identificar funcionalidades .....	22
2.2.2 Descripción de las funcionalidades.....	25
2.3 Diagrama de paquetes .....	30
2.4 Patrones de diseño .....	31
2.5 Conclusiones parciales .....	31
Capítulo 3. Desarrollo y prueba de la solución propuesta .....	33
3.1 Planificación de la implementación.....	33
3.2 Tareas de Ingeniería.....	33
3.3 Estándares de codificación .....	35
3.4 Pruebas de sistema .....	37
3.4.1 Pruebas de integración.....	37
3.4.2 Pruebas de aceptación .....	40
3.5 Conclusiones parciales .....	42
Conclusiones .....	43
Referencias bibliográficas .....	44
ANEXOS.....	47

## Índice de figuras

<i>Figura 1: Arquitectura de Nova NAS.</i> .....	14
<i>Figura 2: Interfaz escanear.</i> .....	16
<i>Figura 3: Interfaz configuraciones.</i> .....	16
<i>Figura 4: Diagrama de paquetes.</i> .....	30
<i>Figura 5: Integración del módulo al sistema.</i> .....	38
<i>Figura 6: Integración del módulo al sistema_licencia.</i> .....	39
<i>Figura 7: Integración del módulo al sistema_actualización.</i> .....	39
<i>Figura 8: Casos de prueba.</i> .....	41

## Índice de tablas

<i>Tabla 1: Comparación entre los antivirus Kaspersky y SAV Unix.</i>	9
<i>Tabla 2: Comparación entre los antivirus Kaspersky y SAV Unix según las 4S.</i>	12
<i>Tabla 3: Requisitos funcionales y no funcionales (LRP).</i>	23
<i>Tabla 4: Adicionar licencia.</i>	26
<i>Tabla 5: Configurar actualización.</i>	27
<i>Tabla 6: Escaneo de directorio.</i>	28
<i>Tabla 7: Adicionar la licencia para SAV Unix.</i>	33
<i>Tabla 8: Configurar la actualización para SAV Unix.</i>	34
<i>Tabla 9: Escanear un directorio con SAV Unix.</i>	34
<i>Tabla 10: Planificación de las pruebas.</i>	40
<i>Tabla 11: Caso de prueba ruta correcta para la licencia.</i>	47
<i>Tabla 12: Caso de prueba ruta incorrecta para la licencia.</i>	48
<i>Tabla 13: Caso de prueba ruta con fecha incorrecta para la licencia.</i>	49
<i>Tabla 14: Caso de prueba ruta vacía para la licencia.</i>	49
<i>Tabla 15: Caso de prueba ruta correcta para actualizar.</i>	50
<i>Tabla 16: Caso de prueba ruta incorrecta para actualizar.</i>	51
<i>Tabla 17: Caso de prueba campo Servidor o Ip vacío en el proxy.</i>	51
<i>Tabla 18: Caso de prueba campo Puerto vacío en el proxy.</i>	52
<i>Tabla 19: Caso de prueba campo Usuario vacío en el proxy.</i>	52
<i>Tabla 20: Caso de prueba campo Contraseña en el proxy.</i>	53
<i>Tabla 21: Caso de prueba valores incorrectos en el proxy.</i>	54
<i>Tabla 22: Caso de prueba escanear directorio.</i>	54
<i>Tabla 23: Caso de prueba escaneo programado de directorio.</i>	55
<i>Tabla 24: Caso de prueba escaneo programado de directorio no activado.</i>	56

## Introducción

La constante evolución de las Tecnologías de la Información y la Comunicación (TIC) ha tenido un impacto positivo en la vida del hombre y ha incidido de forma directa o indirecta en la mayoría de los sectores de la sociedad. Estas proporcionan nuevas formas de acceder, transmitir, almacenar, buscar o mostrar información. Contando con el factor de ubicuidad (sin importar el lugar y la hora para su uso) traen consigo nuevas visiones para el desarrollo nacional. Cuba se enfrenta a la necesidad de migrar a software libre para lograr la soberanía tecnológica, así como informatizar el país dado a las condiciones políticas, económicas, sociales y tecnológicas que impulsan el desarrollo. Con la misión de informatizar la sociedad en dirección a su desarrollo, eficiencia en sus procesos y con mejor calidad de vida, así como lograr el perfeccionamiento funcional de los Organismos de la Administración Central del Estado (OACE), ha puesto en marcha diferentes proyectos.

La Universidad de las Ciencias Informáticas (UCI) fue creada con dos objetivos, informatizar el país y desarrollar la industria del software para contribuir al desarrollo económico del mismo, con la misión de: formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática; producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación; servir de soporte a la industria cubana de la informática [1]. En dicha universidad se encuentra el Centro de Software Libre (CESOL) en el cual se desarrollan proyectos con dirección al proceso de migración a software libre, resaltando entre estos el primer sistema operativo de producción nacional Nova, con variantes para escritorio (Nova Escritorio), servidores (Nova Servidor) y para escritorios ligeros (Nova Ligero).

Nova NAS es una personalización de Nova Servidor para la administración y configuración de un servidor de almacenamiento en la red (NAS, del inglés *Network Attached Storage*) con las principales tecnologías de administración de almacenamiento. Dentro de estas tecnologías se encuentra la gestión de redundancia de discos (RAID, del inglés *Redundant Array of Independent Disks*), la gestión de volúmenes lógicos (LVM, del inglés *Logical Volume Manager*), los sistemas de archivos y monitoreo del estado de discos duros (SMART, del inglés *Self Monitoring Analysis and Reporting Technology*), la configuración de servicios de disponibilidad de información, la compartición de archivos (SMB, del inglés *Server Message Block*), la administración remota (SSH, del inglés *Secure SHell*), la compartición en la red (NFS, del inglés *Network File System*), la transferencia de archivos (FTP, del inglés *File*

*Transfer Protocol*), el protocolo de transferencia de ficheros (TFTP, del inglés *Trivial File Transfer Protocol*) y el protocolo de acceso de audio digital (DAAP, del inglés *Digital Audio Access Protocol*). El propósito principal de Nova NAS es brindar un servicio de almacenamiento que garantice la seguridad y la disponibilidad de los datos en los OACE. Esta es una aplicación web que gestiona los servicios y herramientas de forma centralizada, soporta multilinguaje y provee una interfaz intuitiva.

La protección contra el accionar de los programas malignos es vital en todo sistema informático para mantener la confidencialidad, disponibilidad e integridad de los datos; por lo que todo sistema debe estar acompañado de un antivirus. Los antivirus son programas informáticos que se encargan de detectar y eliminar los diferentes virus informáticos. Los virus no solo se extienden a través de ordenadores Windows, cada día cientos de miles de servidores web son infectados en todo el mundo aprovechando vulnerabilidades en el software o fallos de seguridad en el código de las aplicaciones web instaladas en esos servidores [2]. Actualmente Nova NAS realiza el escaneo y detección de programas maliciosos en los datos con el antivirus ClamAV. Según la Resolución No. 127/2007 del Ministerio de la Informática y las Comunicaciones (MIC), documento legal que establece los requerimientos de seguridad informática, se utilizarán los programas antivirus de producción nacional u otros autorizados oficialmente para usarse en el país [3]. ClamAV no es nacional y su uso no está autorizado por lo que se quebranta dicha resolución tomando un camino opuesto a la política de Cuba. La ausencia de ClamAV supondría no tener defensa contra los virus, o sea, una brecha para el acceso a la información por parte de los programas malignos, por lo que los datos no serían seguros y Nova NAS no sería funcional.

Durante el proceso de investigación sobre los programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país, se ha planteado la siguiente interrogante que define el **problema científico**: ¿Cómo garantizar la administración de un antivirus en Nova NAS que cumpla con la Resolución No. 127/2007 del MIC?

Con el propósito de brindarle una solución efectiva al problema, se plantea como **objetivo general** desarrollar un módulo para Nova NAS que permita la administración de un antivirus que cumpla con la Resolución No. 127/2007 del MIC.

El **objeto de estudio** de dicha investigación lo constituye la administración de programas antivirus, centrado su **campo de acción** en la administración de programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país.

En el marco de esta investigación se ha desglosado el objetivo general en los siguientes **objetivos específicos**:

- Caracterizar los programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país.
- Analizar y diseñar un módulo para la administración de antivirus en Nova NAS.
- Implementar el módulo para la administración de antivirus en Nova NAS.
- Realizar las pruebas correspondientes al módulo implementado.

Para darle cumplimiento a los objetivos trazados se definen las siguientes **tareas de investigación**:

- Caracterización de los programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país.
- Definición y especificación de requisitos.
- Descripción de la arquitectura del módulo.
- Implementación del módulo propuesto.
- Planificación y realización de los casos de prueba al módulo implementado.

En la presente investigación se plantea como **idea a defender** que el desarrollo de un módulo que integre un antivirus que cumpla con la Resolución No. 127/2007 del MIC permitirá su administración desde Nova NAS.

La investigación realizada como parte del trabajo se sustenta en el empleo de los **métodos científicos** siguientes: el Análisis Histórico - Lógico para determinar los antecedentes históricos relacionados con módulos para Nova NAS que integren antivirus; Analítico – Sintético en la investigación de los antivirus autorizados en el país y sus componentes, ayuda a escoger los más apropiados para ser utilizados en el módulo para Nova NAS. La entrevista es una técnica de recolección de información, permite la recolección de datos relacionados con Nova NAS y los antivirus autorizados para su uso en Cuba.

## Introducción

El presente documento se encuentra dividido en tres capítulos. En el primero de ellos, Fundamentación teórica del módulo antivirus para Nova NAS, se realiza un análisis de programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país. Además de compararlos y seleccionar uno para el módulo que será integrado a Nova NAS. También se definen los requisitos de Nova NAS para integrarle un módulo, así como las herramientas y metodología a emplear.

En el segundo capítulo, Análisis y diseño del módulo para Nova NAS, se presenta la fase de Planificación-Definición definida por la metodología SXP para cumplir objetivos formulados y se exponen los requisitos funcionales y no funcionales del sistema propuesto. También se realizan las descripciones de las Historias de Usuario, se explican los patrones de diseño empleados y un análisis de la arquitectura (Diagrama de Paquetes).

En el tercer capítulo, Desarrollo y prueba de la solución propuesta, se implementa dicha solución, mostrando las Tareas de Ingeniería y la Planificación de Liberación. Así mismo se confecciona el plan de pruebas y la ejecución de las pruebas necesarias para verificar que las funcionalidades desarrolladas dan cumplimiento a los requisitos planteados. Se explican los resultados obtenidos y el aporte del trabajo.

## Capítulo 1. Fundamentación teórica del módulo antivirus para Nova NAS

Con el objetivo de facilitar la comprensión del alcance de la investigación, en el presente capítulo se exponen conceptos fundamentales asociados al dominio del problema planteado. Además, se realiza un análisis detallado del estado del arte que precede a la realización de este trabajo y que contribuye a esclarecer su objeto de estudio. Como elemento adicional se exponen las principales características de SAV Unix, Nova NAS, así como las herramientas a emplear, tecnologías y metodología de desarrollo con sus descripciones.

### 1.1 Conceptos Fundamentales

#### 1.1.1 Virus Informáticos

En el libro Redes, Comunicaciones y el Laboratorio de Informática los autores Yáñez y García se refieren a virus informáticos como: "...programas diseñados intencionalmente con el propósito de dañar o destruir la información en los ordenadores. Existen distintos tipos de estos programas destructores, siendo los virus sólo uno de ellos. Estas clasificaciones de los programas destructores se basan en su forma de actuar, y entre estos tipos se encuentran los llamados gusanos (*worms*), Caballos de Troya, los goteros (*droppers*), los virus propiamente dichos y otros. Tienen en común que actúan sin conocimiento del usuario y el daño que causan dentro de la computadora: alteran o borran información, formatean discos, disminuyen considerablemente la velocidad de procesamiento y en algunos casos ocasionan daños en el hardware..." [4].

En este sentido como conceptos asociados a virus informáticos los autores Yáñez y García determinaron [4]:

**Gusanos:** son programas destructores que funcionan en grandes sistemas informáticos conectados mediante una red de comunicaciones y se difunden rápidamente a través de esta. Estos programas hacen una gran utilización de la memoria del sistema, por lo que provocan una disminución en la velocidad de procesamiento y bloquean el funcionamiento del sistema. Son capaces de replicarse.

**Caballos de Troya:** son programas destructores conocidos así, porque su mecanismo de acción es similar al utilizado por los griegos para entrar en Troya (las tropas griegas estaban escondidas dentro de un caballo gigantesco aparentemente ofrecido como regalo o abandonado). Estas tropas



aguardaron a que los troyanos entraran el caballo a su ciudad y en determinado momento salieron y la tomaron. En los Caballos de Troya el programa destructor está escondido dentro de un programa inofensivo en apariencia, que al ejecutarse provoca la destrucción.

**Virus:** los virus informáticos son los máximos exponentes de los programas destructores. Son capaces de reproducirse a sí mismos sin que el usuario esté consciente de lo que sucede. Se adicionan a programas de aplicación, así como a componentes del sistema operativo, de forma tal que pueden tomar el control del mismo durante la ejecución del programa infectado. El código del virus se ejecuta antes que el programa original, y una vez que haya realizado la acción para la que fue diseñado, le da el control al programa, con el objetivo de que el usuario no note su presencia (a este proceso se llama enmascaramiento del virus). Son los tipos de programas malignos que más abundan y más daño causan, y sobre los cuales hay mayor conocimiento.

### 1.1.2 Antivirus

Se denomina antivirus a un software utilizado para eliminar programas elaborados con intención destructiva. Así, los antivirus surgieron como una solución a la proliferación de software malicioso cuando el uso de computadoras personales comenzó a masificarse y con ello surgió todo un nuevo mercado [5].

Existen una gran variedad de antivirus con diferentes funciones y formas de manejarlos, pero constan de características que hacen a estos más seguros para un ordenador, tales como [6]:

- Capacidad de detección de virus.
- Capacidad de eliminación de infecciones.
- Capacidad de actualización de las bases de datos para detectar nuevos virus.
- Integración con el correo electrónico.
- Capacidad de creación de copias de seguridad o *back-ups*.
- Capacidad de detección de otros tipos de malware y peligros como *Spam*, *spyware*, *phishing*, etc.
- Servicio de atención al cliente y apoyo técnico.

Entre las formas más conocidas de clasificar los antivirus se encuentran [7]:

- **Antivirus Preventores:** como su nombre lo indica, este tipo de antivirus se caracteriza por anticiparse a la infección, previniéndola. De esta manera, permanecen en la memoria de la computadora, monitoreando ciertas acciones y funciones del sistema.
- **Antivirus Identificadores:** esta clase de antivirus tiene la función de identificar determinados programas infecciosos que afectan al sistema. Los antivirus identificadores también rastrean secuencias de bytes de códigos específicos vinculados con dichos virus.
- **Antivirus Descontaminadores:** comparten una serie de características con los identificadores, sin embargo, su principal diferencia radica en el hecho de que el propósito de esta clase de antivirus es descontaminar un sistema que fue infectado, a través de la eliminación de programas malignos. El objetivo es retornar dicho sistema al estado en que se encontraba antes de ser atacado.

En Cuba el uso de antivirus está regido por la Resolución No. 127/2007 del MIC, documento que establece un reglamento de seguridad para las TIC.

## 1.2 Análisis de la Resolución No. 127/2007 del MIC

Con el creciente desarrollo de la informatización de los procesos de la sociedad se hace evidente la necesidad de reglamentos que vayan en dirección al fortalecimiento y duración de los mismos. Además, se requiere la adopción de medidas que garanticen un adecuado nivel de seguridad para su protección y ordenamiento.

En Cuba la seguridad informática está amparada legalmente por la Resolución No. 127/2007 del MIC, la cual pone en vigor el reglamento de seguridad en las tecnologías de la información. La misma tiene por objetivo establecer los requerimientos que rigen la seguridad de las tecnologías de la información, garantizando un respaldo legal que responda a las condiciones y necesidades de la informatización del país. Además, la resolución no sustituye las medidas específicas que norman el procesamiento de la información clasificada y limitada emitidas por el Ministerio del Interior.

El capítulo 3, Empleo conveniente y seguro de las tecnologías de la información, está compuesto por 8 secciones las cuales tienen sus particularidades. En la sección 6, Seguridad ante programas malignos, el artículo 50 regula el uso de los antivirus, el mismo plantea:

“En cada entidad se implementarán los controles y procedimientos para protegerse contra virus y otros programas dañinos que puedan afectar los sistemas en explotación, así como para impedir su generalización. Para la protección contra virus se utilizarán los programas antivirus de producción nacional u otros autorizados oficialmente para su uso en el país, debidamente actualizados” [3].

Consecuente con lo planteado en dicha resolución, a continuación, se muestra un estudio de los antivirus de producción nacional u otros autorizados oficialmente para su uso en el país.

### **1.3 Antivirus de producción nacional u otros autorizados oficialmente para su uso en el país**

En la actualidad existen diversos antivirus que funcionan sobre sistemas operativos GNU/Linux, haciéndolos cada vez más seguros. Con la búsqueda detallada de los antivirus que cumplan con lo planteado en la Resolución No. 127/2007 del MIC y la realización de una entrevista (Ver Anexo 1) al Ing. Juan Carlos Casas, especialista del Grupo Soporte Técnico de la Empresa Segurmática, empresa especializada en consultoría y seguridad informática, la cual es la encargada de comercializar y desarrollar diversos productos antivirus tanto para Linux y Windows, se determinaron los antivirus nacionales e internacionales autorizados para su uso en el país.

#### **1.3.1 Análisis de la entrevista**

La producción nacional de antivirus está guiada por la empresa Segurmática; consta de diversos productos donde solo para Linux se encuentra SAV Unix. Dentro de los antivirus internacionales oficialmente autorizados se encuentra Kaspersky como único candidato, puesto que existen relaciones contractuales con Kaspersky Lab (empresa donde se desarrolla Kaspersky), la cual permite la venta en Cuba de forma legal de su licencia.

#### **1.3.2 Antivirus de producción nacional**

De SAV Unix se pudo constatar que es un [8]: programa antivirus capaz de detectar programas malignos que se ejecutan en Windows y Linux. Está diseñado para la consola de Linux y recibe parámetros por el *prompt* de comandos. El método de actualización es diferencial, de manera predeterminada desde el sitio:

<http://www.segurmatica.cu/descargas/savunix/linux>.

### 1.3.3 Antivirus internacionales oficialmente autorizados

En el caso de Kaspersky para servidores de Linux se tiene que [9]:

Ofrece protección de varias capas para servidores de archivos en Linux y redes heterogéneas que funcionan simultáneamente en el nivel del sistema de archivos y en el nivel del protocolo de transferencia de datos SMB/CIFS (servidor Samba). El principal módulo antivirus (un interceptor en el nivel de *kernel*) protege el sistema de archivos del servidor en tiempo real. La protección se extiende tanto a los recursos locales como a los recursos remotos integrados en el sistema de archivos del servidor al que se puede obtener acceso mediante una variedad de protocolos de transferencia de datos.

Consta de diferentes características [9]:

Protección contra software malicioso cuando la aplicación se reinicia automáticamente si ocurre un mal funcionamiento o debe finalizar.

Puede utilizarse para proporcionar una potente protección contra software malicioso a las redes con los sistemas operativos menos convencionales.

### 1.3.4 Comparación de los antivirus

Se determinó SAV Unix y Kaspersky como los antivirus que cumplen con la Resolución No. 127/2007 del MIC y pudiesen ser integrados a Nova NAS en un módulo. Como precedente a este trabajo, en el año 2013 los autores Rosniel Arian Montanet Baños y Marisé Bonilla Marzá propusieron en su trabajo de diploma un módulo para la administración de SAV Unix en Nova Servidor, después de comparar SAV Unix y Kaspersky. A continuación, se cita la tabla comparativa que ellos realizaron entre dichos antivirus.

Tabla 1: Comparación entre los antivirus Kaspersky y SAV Unix [10].

Características	Kaspersky	SAVUnix
Valoración		
Calidad	Muy Bien	Muy Bien

Calidad Precio	Bien	Muy Bien
<b>Puntuación</b>		
Detección en análisis	Muy Bien	Muy Bien
Detección en tiempo real	Muy Bien	-
Velocidad análisis	Muy Bien	Bien
Falsas amenazas detectadas	Excelente	-
Rendimiento del equipo	Bien	Muy Bien
Interfaz	Muy Bien	-
<b>Certificados independientes</b>		
AV COMPARATIVES	Sí	-
VIRUS BULLETIN	No	-
ICSA LABORATORIES	Sí	-
WEST COAST LABS ANTIVIRUS	Sí	-
<b>Protección detallada</b>		
Anti-virus	Sí	Sí
Anti-troyanos	Sí	Sí
Anti-spyware	Sí	Sí
Anti-gusanos	Sí	Sí
Anti-phishing	Sí	Sí
Anti-rootkits	Sí	Sí

Protección Email/Pop3	Sí	Sí
Protección P2P	Sí	Sí
Protección mensajería instantánea	Sí	Sí
<b>Compatibilidad</b>		
Windows 8	Sí	Sí
Windows 7 Vista	Sí	Sí
Windows XP	Sí	Sí
MAC OS	Sí	No
Linux	Sí	Sí
<b>Actualizaciones</b>		
Automática	Sí	Sí
Manual	Sí	Sí
Frecuencia de actualizaciones	24 horas	24 horas
Protección del registro de sistema	Sí	Sí
Auto-limpieza de infecciones	Sí	Sí
Cuarentena para infecciones	Sí	-
Modo juego	Sí	-
Autodetección USB	Sí	Sí
Navegación segura	Sí	-

Comparados estos se concluye que ambos son excelentes antivirus y poseen características y funcionalidades óptimas para lograr la seguridad necesaria en Nova NAS.

## 1.4 Selección del antivirus

Desde el año 2010 el desarrollo de los productos de Nova están dirigidos al cumplimiento de los principios para el desarrollo, uso y aplicación de las TIC en los OACE, también llamados las 4S (Soberanía, Socio-Adaptabilidad, Seguridad, Sostenibilidad). Estos principios son utilizados en la práctica diaria para el análisis y selección de distintas tecnologías, teniendo en cuenta que en caso de tener varias alternativas siempre se opta por las que más afines sean con las 4S [10].

Debido a que el uso de Kaspersky en Cuba está determinado por las relaciones contractuales existentes entre Segurmática y Kaspersky Lab, y la ruptura de dicha relación fuese un problema para el uso de dicho antivirus, además de poner en riesgo la soberanía tecnológica usando un antivirus de producción internacional, se concluyó como más indicado para integrar el módulo en Nova NAS a SAV Unix. En la Tabla 2 se muestra la comparación que hacen Montanet Baños y Bonilla Marzá teniendo en cuenta el cumplimiento de las 4S.

Tabla 2: Comparación entre los antivirus Kaspersky y SAV Unix según las 4S [10].

<b>Características</b>	<b>Kaspersky</b>	<b>SAV Unix</b>
Principios para el desarrollo, uso y aplicación de las TIC en los OACE		
<b>Soberanía</b>	No	Sí
<b>Socio-Adaptabilidad</b>	No	Sí
<b>Seguridad</b>	No	Sí
<b>Sostenibilidad</b>	Sí	Sí

Esta selección se basó también en cinco aspectos o pilares fundamentales, los cuales se exponen a continuación:

- Utiliza una base de datos de virus que circulan dentro de Cuba, lo que proporciona además una seguridad adicional para virus nacionales que Kaspersky no posee.
- El método de actualización es fácil, asequible y diferencial, de manera predeterminada desde el sitio <http://www.segurmatica.cu/descargas/savunix/linux>.

- Es de producción nacional, lo que facilita su soporte e información.
- Amplia compatibilidad tecnológica (Linux, RedHat, SuSE, Mandriva, Debian, Ubuntu, Nova, Fedora, etc.).
- Es tanto para servidores como estaciones de trabajo en Linux.

Una vez determinado SAV Unix como antivirus a administrar, se deben tener en cuenta las características y requisitos de Nova NAS, para integrarle SAV Unix en un módulo.

## 1.5 Descripción de Nova NAS

Consiste en una personalización de una distribución cubana de GNU/Linux orientada a brindar servicio de almacenamiento conectado a la red (NAS). Como bien plantea Villazón esta es: "...una aplicación web para la administración de un servidor de almacenamiento en la red (NAS) desplegada con Nova Servidor 2015 que gestiona los servicios y herramientas de forma centralizada y provee una interfaz intuitiva..." [11]. Dentro de sus características más relevantes se encuentra su administración la cual es basada en la red, contiene listas de control de acceso (ACL, del inglés *Access Control List*), cuenta con un soporte multilinguaje y posee licencia GPL.

### 1.5.1 Conceptos asociados

**NAS:** tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento en un servidor con computadoras personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un sistema operativo optimizado para dar acceso con los protocolos SMB, NFS o FTP [11].

**Nova Servidor 2015:** distribución GNU/Linux cubana orientada a servidores basado en Ubuntu Server. Es un sistema optimizado para reducir al máximo el consumo de los recursos de hardware (no tiene interfaz gráfica) [11].

### 1.5.2 Arquitectura

Nova NAS consta de una arquitectura en 2-Capas. En la Figura 1 se muestran las capas que componen la arquitectura, las cuales se describen a continuación:



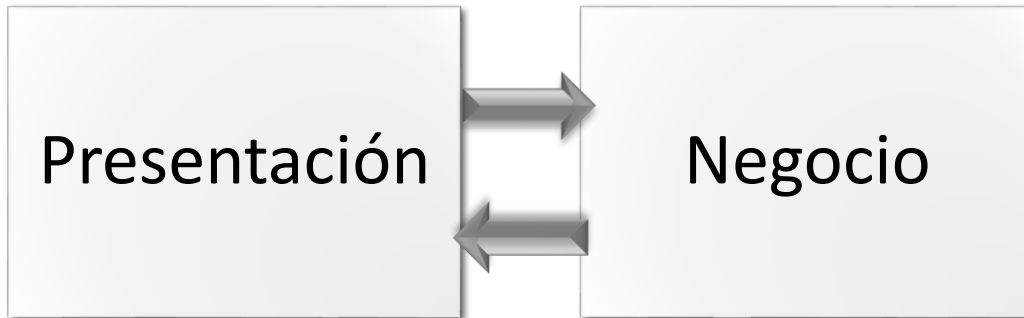


Figura 1: Arquitectura de Nova NAS.

- Capa de Presentación: es la encargada de presentar al usuario los conceptos de negocio mediante una interfaz de usuario (UI, del inglés *User Interface*), facilitar la explotación de dichos procesos, informar sobre la situación de los procesos de negocio e implementación de las reglas de validación de dicha interfaz. Permite interactuar con la aplicación. Se comunica con la capa de Negocio. Presenta todos los servicios que pueden ser administrados desde Nova NAS, así como diferentes opciones de configuración, seguridad, diagnóstico e información. Es desarrollada mediante Ext JS.
- Capa de Negocio: es donde son procesados los datos. Recibe y procesa peticiones del usuario. Establece las reglas del negocio. Emite respuestas a la capa de Presentación. Tiene asignada la responsabilidad de contener el código necesario para el manejo de datos. Es desarrollada mediante PHP.

### 1.5.3 Requisitos o consideraciones para integrarle un módulo

Un módulo brinda nuevas funcionalidades, así como contenido específico. Según el Diccionario de Informática y Tecnología un módulo es: "...un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo..." [12].

En el caso de Nova NAS el módulo debe cumplir con la arquitectura antes mencionada (Ver epígrafe 1.5.2) integrándose a cada una de sus capas, así como con el uso correcto de las tecnologías y herramientas con que fue desarrollado Nova NAS.

## 1.6 Administración de antivirus

Generalmente, el proceso de instalación de un antivirus se realiza mediante una UI que permite llevar a cabo el proceso de administración de las configuraciones del mismo. Los antivirus requieren una correcta configuración para funcionar adecuadamente. Adicionar una licencia es la primera configuración que se debe realizar, la cual tiene la función de acreditar y validar el uso del antivirus. La configuración de la actualización permite establecer la forma en que se actualizará el antivirus para mantener la seguridad en correspondencia con los nuevos virus que son identificados.

### 1.6.1 Administración del servicio antivirus SAV Unix

SAV Unix es del tipo descontaminador, puesto que su propósito es descontaminar un sistema que fue infectado a través de la eliminación de programas malignos y retornar dicho sistema al estado en que se encontraba antes de ser atacado. Este no posee una interfaz gráfica de usuario, su administración es mediante la consola, o sea, comprende diferentes rutas y comandos que llevan a su correcta actualización, soporte y funcionamiento. El dominio de las configuraciones es importante para administrar y permitir el correcto funcionamiento del antivirus en su totalidad. A continuación, se muestran ficheros de configuración, directorios de instalación, así como comandos importantes.

Para instalar SAV Unix desde un repositorio se ejecuta el comando ***apt-get install sav-unix***. También se pudiese descargar el programa de un sitio web que lo facilite y ejecutar ***dpkg -i Sav\_Unix\_2.6-1\_i386.deb*** o ***rpm -i SAV\_Unix-2.6-1.i586.rpm*** para su instalación. El directorio principal donde se encuentra dicho antivirus es: ***/opt/sav***. Para acceder al fichero de configuración se ejecuta en la consola ***sudo /opt/sav/bin/config.sh***, el cual muestra un script que guía a través del proceso de configuración, donde las opciones son: ver información de la licencia, adicionar o cambiar la licencia, ver la configuración del actualizador y modificarla. SAV Unix consta de los comandos ***sscanner*** para la búsqueda y descontaminación de códigos malignos y ***savupd*** para realizar la configuración y descarga de la actualización.

Teniendo en cuenta que el uso de la consola es considerado por muchos usuarios un elemento negativo y complicado debido a la necesidad de conocimientos avanzados en el empleo de comandos, han surgido diferentes proyectos dirigidos al desarrollo de una interfaz gráfica de usuario para la administración del servicio antivirus SAV Unix. En la investigación de sistemas que permitan administrar SAV Unix se encuentra la interfaz desarrollada por el Ing. Alexander Martínez Fajardo de CESOL la cual permite su administración (actualizar, escanear, adicionar la licencia, seleccionar la dirección de las actualizaciones y configurar el proxy) (Ver Figuras 2 y 3).



Figura 2: Interfaz escanear.



Figura 3: Interfaz configuraciones.

Partiendo de que Nova NAS cuenta con una interfaz web para administrar sus servicios con una arquitectura 2-Capas (Ver epígrafe 1.5.2) y que la UI antes descrita no cumple con dicha arquitectura, se determina que no puede ser integrada a Nova NAS. Por tal motivo es que se hace necesario desarrollar un módulo para administrar SAV Unix desde Nova NAS.

## 1.7 Herramientas y tecnologías a emplear

Para el desarrollo del módulo propuesto se definió el uso de las siguientes herramientas y tecnologías, estas desempeñan un importante papel en el desarrollo del módulo, debido a que Nova NAS está desarrollada con las mismas.

### 1.7.1 Lenguajes de programación utilizados para el desarrollo del módulo

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo [13].

#### **PHP** (del inglés *Hypertext Preprocessor*)

Es un lenguaje interpretado de alto nivel embebido en el lenguaje de marcas de hipertexto (HTML, del inglés *HyperText Markup Language*) y ejecutado en el servidor [14]. Este es utilizado para el desarrollo de la capa de Negocio, para el manejo y procesamiento de datos.

Generalmente los scripts en PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas web enormemente. La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente (un navegador que solicita una página web) sólo recibe el resultado de la ejecución y jamás ve el código PHP [15].

#### **BASH**

Es un shell de Unix (intérprete de comandos de Unix) escrito para el proyecto GNU. Su nombre es un acrónimo de bourne-again shell (otro shell bourne); haciendo un juego de palabras (born-again significa renacimiento) sobre el Bourne shell (sh), que fue uno de los primeros shells importantes de Unix [16].

El shell actúa como un intermediario entre el sistema operativo y el usuario utilizando líneas de comando introducidas por dicho usuario. Su función consiste en la lectura de la línea de comandos, la

interpretación de su significado, la ejecución del comando y luego la devolución del resultado a través de las salidas [17]. El mismo se utiliza en la capa de Negocio para la ejecución de comandos.

### **1.7.2 Framework de desarrollo**

Un framework de desarrollo es un software que ofrece una infraestructura para la creación de otros programas. El framework contiene librerías de código y módulos ya listos que resumen las tareas de creación de elementos recurrentes en el desarrollo de aplicaciones, a la vez que define una arquitectura para el desarrollo de software [18].

#### **Ext JS**

Es una librería Javascript que permite construir aplicaciones complejas en Internet. Esta librería incluye [19]:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias *Open Source* y comerciales.
- Permite el desarrollo de interfaces en la capa de Presentación. Comunica y captura información.

### **1.7.3 Entorno de Desarrollo Integrado (IDE)**

Un Entorno de Desarrollo Integrado (IDE, del inglés Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Los componentes de cualquier entorno de desarrollo integrado son un editor de texto, un compilador, un intérprete, un depurador, que tenga posibilidad de ofrecer un sistema de control de versiones y que ayude en la construcción de interfaces gráficas de usuario [20].

#### **PhpStorm**

Es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre. Estas son sólo algunas de las características principales [21]:

- Permite la gestión de proyectos fácilmente.
- Proporciona un fácil autocompletado de código.
- Soporta el trabajo con PHP 5.5.
- Sintaxis abreviada.

#### **1.7.4 Herramienta CASE para el modelado**

Son necesarias para aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

##### **Visual Paradigm para UML**

Es una herramienta de Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Language*) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos [22]. Esta herramienta de fácil uso permite con el modelado visual UML un desarrollo rápido y aceptado del ciclo de vida completo de software.

#### **1.8 Metodología de desarrollo utilizada**

Las metodologías de desarrollo de software proporcionan un camino más fácil y seguro para el desarrollo, siendo estas un conjunto de procedimientos, técnicas, herramientas y un soporte documental, que ayuda a los desarrolladores a realizar nuevo software. La metodología define roles los cuales indican quién debe hacer qué, cuándo y cómo debe hacerlo para obtener los distintos productos parciales y finales. Se clasifican en dos tipos: tradicionales y ágiles.

Como metodología de desarrollo se utilizó SXP (SCRUM-Extreme Programming) debido a que el equipo de desarrollo posee un amplio dominio de la misma y Nova NAS ha sido desarrollado mediante esta, lo que hace del proceso de desarrollo más familiar para el cliente al igual que los artefactos a utilizar.

Como bien plantea Peñalver, es un híbrido cubano que tomando las mejores prácticas de SCRUM del inglés *scrum* y XP (*Extreme Programing*), parte de procedimientos ágiles para realizar estrategia tecnológica que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Con una programación rápida o extrema, incorpora como parte del equipo, al usuario final, pues no hay mayor seguridad de que se lleva el proyecto en la dirección correcta siendo esto uno de los requisitos para llegar al éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y consta de métricas o indicadores para lograr una eficiente calidad [23].

Consta de 4 fases principales [23]:

- Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega, puesta en marcha.
- Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad [23].

### **1.9 Conclusiones parciales**

En el desarrollo del presente capítulo se analizó la Resolución 127/2007 del MIC, permitiendo conocer los antivirus que cumplen con la misma, compararlos y determinar a SAV Unix como el más indicado para ser administrado desde Nova NAS. Además, se realizó una breve descripción del funcionamiento y las principales características de Nova NAS, lo que permitió tomar como punto de partida dichas

## Capítulo 1

consideraciones para concebir la solución al problema planteado, definir herramientas a emplear y la metodología de desarrollo de software que guiará dicho proceso.



## **Capítulo 2. Análisis y diseño del módulo para Nova NAS**

En el presente capítulo como parte del proceso de desarrollo de software se realizará un análisis y diseño, que garantice identificar los requisitos con el objetivo de establecer que debe ser capaz de realizar el módulo antivirus. Son detalladas las funcionalidades, arquitectura y patrones de diseño a utilizar, tomando como punto de partida las características de Nova NAS.

### **2.1 Propuesta del módulo a desarrollar**

Después de realizar un estudio de las principales tecnologías asociadas a los NAS y de las herramientas de administración de este servidor existentes; para darle solución al problema planteado, el presente trabajo de diploma propone desarrollar un módulo que integre y permita administrar el antivirus SAV Unix en Nova NAS, en el cual se tengan las funcionalidades necesarias para el correcto funcionamiento del mismo. Dicho módulo brindará la posibilidad de realizar configuraciones para la actualización del antivirus (ruta de actualización y licencia), permitiendo la actualización, así como el escaneo de directorios y la programación de dicho escaneo en un tiempo determinado. Permitirá además, la visualización de información referente al escaneo.

### **2.2 Características y cualidades**

Es de vital importancia identificar las características y cualidades que debe cumplir el módulo, para lograr un entendimiento común entre el usuario y el proyecto de software. En este caso SAV Unix funciona como un servicio el cual es configurado y utilizado mediante el módulo. La generación de los artefactos necesarios e imprescindibles respalda la documentación de cada uno de los sistemas informáticos, logrando así que no queden sin ser analizados y documentados [23].

#### **2.2.1 Identificar funcionalidades**

Los requerimientos funcionales son la determinación exacta de qué debe ser capaz de hacer el sistema, estos se corresponden con opciones que ejecutará el módulo, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

Durante la captura de requisitos consiguiente con SXP se utiliza el artefacto Lista de Reserva del Producto (LRP), en el cual, se definen las funcionalidades que tendrá el producto en forma de requisitos técnicos y de negocio. Es una lista priorizada y garantiza la organización de los requisitos funcionales y no funcionales, a partir de la prioridad que tengan para el desarrollo del sistema, de igual forma facilita el trabajo en la confección de las Historias de Usuario.

A continuación, se muestra un listado con los requisitos funcionales que debe cumplir el módulo de administración del servicio antivirus para Nova NAS.

Tabla 3: Requisitos funcionales y no funcionales (LRP).

<b>Asignado a</b>	<b>Ítem</b> *	<b>Descripción</b>	<b>Estimación</b>	<b>Estimado por</b>
		<b>Prioridad</b>	<b>Muy Alta</b>	
Lázaro Miguel de las Cagigas Barrios	1	Adicionar la licencia	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	2	Configurar actualización del antivirus	0.1	Analista
		<b>Prioridad</b>	<b>Alta</b>	
Lázaro Miguel de las Cagigas Barrios	3	Actualizar SAV Unix	0.1	Analista
Lázaro Miguel de las Cagigas	4	Agregar directorio a escanear	0.1	Analista

Barrios				
Lázaro Miguel de las Cagigas Barrios	5	Modificar directorio a escanear	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	6	Mostrar listado de directorios a escanear	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	7	Eliminar directorio a escanear	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	8	Escanear directorio	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	9	Notificar resultado del escaneo de directorio	0.1	Analista
Lázaro Miguel de las Cagigas Barrios	10	Mostrar resultado del escaneo de directorio	0.1	Analista
<b>RNF (Requisitos No Funcionales)</b>				
Lázaro	1	Ajustarse al diseño de Nova		

Miguel de las Cagigas Barrios		NAS		
Lázaro Miguel de las Cagigas Barrios	2	Solo el administrador puede acceder a la interfaz web		
Lázaro Miguel de las Cagigas Barrios	3	El administrador permanecerá en el sistema el tiempo que así lo considere		
Lázaro Miguel de las Cagigas Barrios	4	La información y las funcionalidades estarán disponibles y el administrador podrá acceder a ellas en todo momento		
Lázaro Miguel de las Cagigas Barrios	5	Las PC clientes deben tener instalado un navegador web		

### 2.2.2 Descripción de las funcionalidades

En la metodología SXP, las Historias de Usuario (HU) constituyen la técnica utilizada con el propósito de especificar los requisitos del software, el equivalente a los casos de uso en el Proceso Unificado. Sirven de guía en la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos

de desarrollo. En este sentido, solo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarla [23].

A continuación, se mostrarán las HU referentes a los requisitos funcionales descritos con anterioridad.

Tabla 4: Adicionar licencia.

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Nombre Historia de Usuario:</b> Adicionar licencia.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lázaro Miguel de las Cagigas Barrios	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0.2
<p><b>Descripción:</b> Una vez dentro de Nova NAS en la parte para la administración de servicios (menú lateral izquierdo) al seleccionar SAV Unix se permitirá adicionar la licencia del antivirus o cambiarla por otra. La aplicación le dará la opción al usuario de elegir la ruta de la licencia (en la pestaña de Licencia).</p> <ul style="list-style-type: none"> <li>• Adicionar licencia: Se introduce la ruta que posee el archivo de licencia en el ordenador con extensión .lic (campo obligatorio). Es introducida mediante el explorador de ficheros (botón Examinar) el cual valida que posea extensión .lic en caso contrario lanza un mensaje de: "Archivo no válido". A su vez es validada para ver si está en fecha de uso, se compara la fecha actual con la fecha de expiración de la licencia (lo realiza internamente el antivirus), si está caduca la licencia, se muestra un mensaje informando al usuario de dicha situación. En caso de no adicionar la licencia no se activa el antivirus, o sea, no es válido su uso.</li> </ul>	

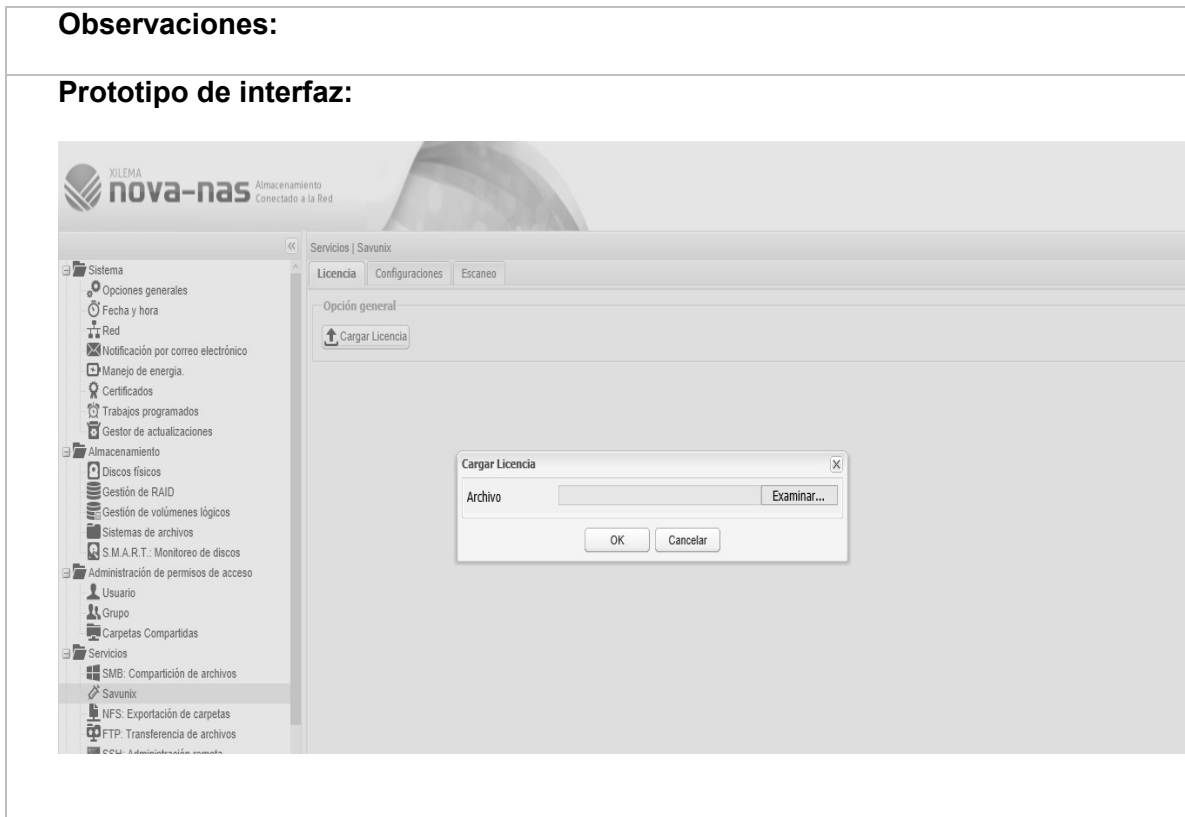


Tabla 5: Configurar actualización.

<b>Historia de Usuario</b>	
<b>Número:</b> 2	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lázaro Miguel de las Cagigas Barrios	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0.2
<b>Descripción:</b> Una vez dentro de Nova NAS en la parte para la administración de	

servicios (menú lateral izquierdo) al seleccionar SAV Unix se permite gestionar la configuración del antivirus (en la pestaña de Configuraciones).

- Establecer ruta de actualización: Permite establecer cómo será la actualización desde un servidor web. Se especifica la dirección del mismo (dicho campo valida que sea en formato URL y no puede estar vacío) y brinda la opción de usar proxy (usuario, contraseña, puerto y servidor o ip) en el mismo se valida que ningún campo este vacío y que cada argumento cumpla con su formato. En caso de dar error en la descarga muestra el mensaje “Error en la descarga”, si el error es por el uso del proxy muestra: “Error de proxy”.
- Actualizar: Una vez establecida la ruta de actualización se brinda la opción de actualizar el antivirus desde la misma (presionando el botón OK).

**Observaciones:**

**Prototipo de interfaz:**

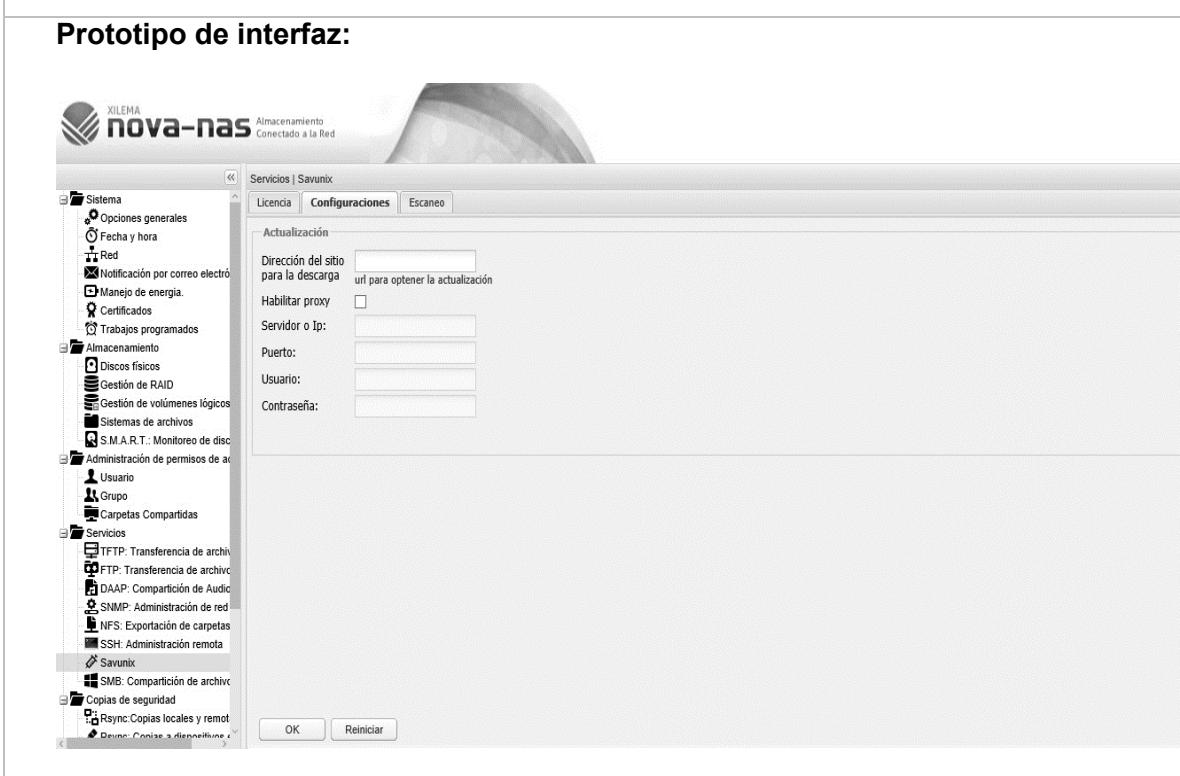
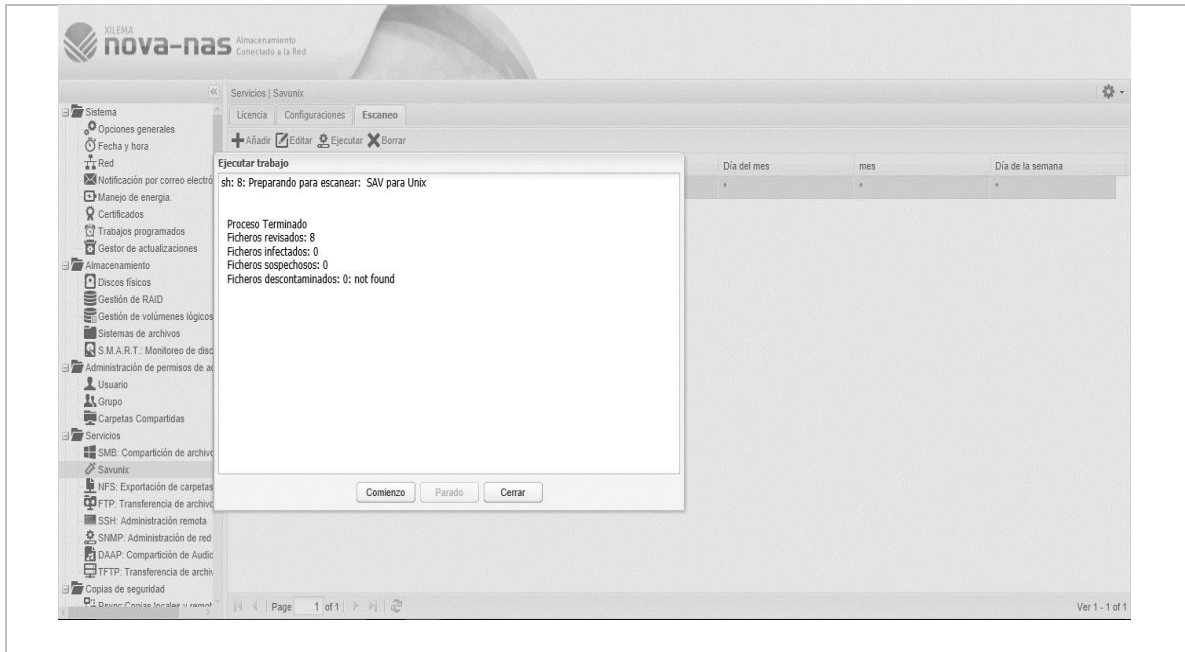


Tabla 6: Escaneo de directorio.

<b>Historia de Usuario</b>	
<b>Número:</b> 3	<b>Nombre Historia de Usuario:</b> Escanear directorio.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Lázaro Miguel de las Cagigas Barrios	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 0.2
<p><b>Descripción:</b> Una vez dentro de Nova NAS en la parte para la administración de servicios al seleccionar SAV Unix (menú lateral izquierdo) se permite agregar, modificar y eliminar un directorio a escanear (en la pestaña Escaneo). Para agregar un directorio a escanear se le dará al usuario la opción de elegir mediante un campo de selección un directorio que se encuentre compartido. Dicho escaneo puede ser ejecutado en tiempo real, seleccionándolo en la lista, presionando el botón Ejecutar (permite realizar un escaneo en cualquier momento) y mostrando su resultado en un mensaje (ficheros revisados, ficheros infectados, ficheros sospechosos y ficheros descontaminados) o programado definiendo para ello el minuto, hora, día del mes, mes y día de la semana. Además, se ofrece la opción de activar o desactivar el escaneo programado y notificar al administrador por correo con dicho resultados.</p>	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	





## 2.3 Diagrama de paquetes

Para mostrar cómo el sistema está dividido en agrupaciones lógicas y sus dependencias se utiliza el diagrama de paquetes. Este muestra las diferentes capas mediante paquetes dentro de los cuales existen subpaquetes que representan cada aplicación del módulo antivirus. A continuación, en la Figura 4, se representan los paquetes relacionados con dicho módulo.

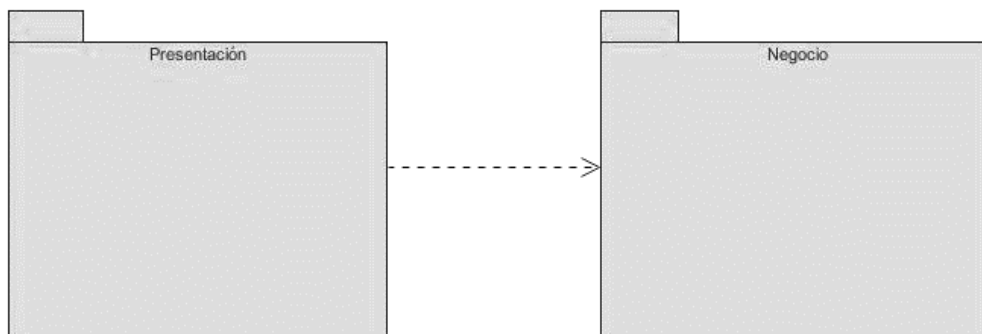


Figura 4: Diagrama de paquetes.

Para la integración de SAV Unix, la capa de Presentación la cual posee código necesario (en el fichero *savunix.js*) para la representación visual del mismo, hace peticiones a la capa de Negocio (a la clase *savunixphp*) que consta del código necesario para el procesamiento de los datos y las peticiones que lleven al correcto funcionamiento de SAV Unix.

### 2.4 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

#### Patrones GRASP

Los GRASP son Patrones Generales de Software para Asignación de Responsabilidades. Aunque se considera que más que patrones propiamente dichos, son una serie de buenas prácticas de aplicación recomendable en el diseño de software. A continuación, son descritos los patrones de este tipo que son utilizados para el diseño de la solución propuesta.

**Experto:** es uno de los más empleados al asignar responsabilidades. Con su empleo, se hizo posible definir dónde colocar en cada clase las funcionalidades que necesitan de esa información, dicha clase sería el experto en información.

Ejemplo: la clase *savunixphp* controla e implementa las funcionalidades que SAV Unix necesita para su funcionamiento (*savunixphp* es experto en esta información).

### 2.5 Conclusiones parciales

El módulo a implementar tuvo un total de 10 funcionalidades, las cuales están organizadas para conformar un total de 3 HU y 5 requisitos no funcionales. La descripción de las HU permitió establecer las capacidades y cualidades con las que el software debe cumplir. Además, la correcta definición de los patrones de arquitectura y de diseño permitieron que se obtuviera una base del sistema capaz de

## Capítulo 2

soportar posteriores cambios en los requisitos. El estudio realizado en este capítulo ha facilitado un entendimiento de la dinámica y estructura de la aplicación, contribuyendo a una mejor comprensión del problema que el módulo a desarrollar debe resolver. Además, se sentaron las bases para las restantes fases del proceso de desarrollo.

## Capítulo 3. Desarrollo y prueba de la solución propuesta

Los procesos de desarrollo de software implican la realización de una serie de actividades que garanticen la calidad del mismo. En el presente capítulo se describen los procesos de implementación y prueba. Se definen las etapas de implementación del sistema, las Tareas de Ingeniería como apoyo al proceso de implementación y se presentan los estándares de codificación empleados para el desarrollo del módulo, para mantener la uniformidad con la codificación de los restantes módulos. Para evaluar la calidad de la solución propuesta se realizan pruebas, estas permiten convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su puesta en funcionamiento.

### 3.1 Planificación de la implementación

Analizadas las HU por parte del cliente, quedando estimado el tiempo y esfuerzo dedicado para desarrollar cada una de ellas, se procede a realizar la planificación de las etapas de implementación del sistema. Este plan concentra las HU por iteraciones, definiendo cuáles serán desarrolladas en cada iteración del proceso de implementación. Teniendo en cuenta lo anteriormente planteado se decide implementar el sistema en una iteración para dar cumplimiento a las HU antes definidas.

### 3.2 Tareas de Ingeniería

Las Tareas de Ingeniería son uno de los artefactos que utiliza la metodología SXP, estas se realizan para facilitar el entendimiento en el proceso de implementación, ya que organizan el mismo con la definición de actividades asociadas a las Historias de Usuarios. A continuación, se definen las Tareas de Ingeniería correspondientes a cada una de las Historias de Usuario especificadas en el capítulo anterior.

Tabla 7: Adicionar la licencia para SAV Unix.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1

<b>Nombre Tarea:</b> Probar los comandos que permitan gestionar la licencia.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 6/03/2016	<b>Fecha Fin:</b> 6/04/2016
<b>Programador Responsable:</b> Lázaro Miguel de las Cagigas Barrios	
<b>Descripción:</b> Se ejecuta en la consola <code>sudo /opt/sav/bin/config.sh</code> , el cual muestra un script que guía a través del proceso de configuración del uso de la licencia.	

Tabla 8: Configurar la actualización para SAV Unix.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Probar los comandos que permitan gestionar la actualización.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 7/03/2016	<b>Fecha Fin:</b> 7/04/2016
<b>Programador Responsable:</b> Lázaro Miguel de las Cagigas Barrios	
<b>Descripción:</b> Se hace uso del comando <code>savupd</code> para realizar la configuración y descarga de la actualización.	

Tabla 9: Escanear un directorio con SAV Unix.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Estudiar el comando que permita escanear un directorio.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 8/03/2016	<b>Fecha Fin:</b> 8/04/2016
<b>Programador Responsable:</b> Lázaro Miguel de las Cagigas Barrios	

**Descripción:** Se ejecuta el comando **sscan**, el cual tiene como finalidad buscar y des-contaminar ficheros infectados o sospechosos.

### 3.3 Estándares de codificación

Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada. La legibilidad del código fuente repercute directamente en la correcta comprensión de un sistema de software [24]. A continuación, se muestran los estándares generales a tener en cuenta para el desarrollo del código PHP, luego de un estudio de la forma en que este se utiliza en Nova NAS.

Etiquetas de apertura y cierre:

- Se utilizarán las etiquetas `<? php y ?>`, y en ningún caso la versión corta `<? y ?>`.

Uso de punto y coma (;):

- Todas las líneas de código se cerrarán con punto y coma.

Asignación de nombres:

- Emplear descriptores en español.
- Evitar nombres largos y que difieran en una letra o en el uso de mayúsculas.
- Para nombrar las funciones y variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se utiliza la notación Camel (escribir palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra a partir de la segunda palabra ejemplo: `cargarServiciosExternos ()`, `$datosInternos`).

Variables:

- Los nombres de las variables deben ser cortos y significativos, siempre acompañados del símbolo \$ (ejemplo: `$datos`).
- La elección de un nombre de variable debe ser mnemotécnica, esto es, diseñado para demostrar el propósito de su uso a cualquier observador.

## Capítulo 3

- Los nombres de variables de un solo carácter deben ser evitados, solo usarlo para variables temporales, como i, j o k para enteros, o c, d o e para caracteres.

### Clases:

- Tratar de que el nombre de la clase sea simple y descriptivo.
- Usar palabras completas, evitar abreviaturas o acrónimos, a menos que sean mucho más usadas que el nombre completo.
- Los nombres de las clases deben ser sustantivos.
- Los nombres de clase deben empezar con una letra mayúscula y el resto de letras deben estar escritas en minúscula. En el caso que pueda tener más de una palabra, las primeras letras de cada palabra interna deben ser con mayúscula (notación CamelCase).

Ejemplo: `class SavUnixControladora { ... }`

### Inicialización:

- Inicializar cada variable en su declaración a menos que su valor inicial dependa de algún cálculo.
- Se debe declarar cada variable en su propia línea.

### Comentarios:

- Antes de una línea o bloque de código.
- Todas las funciones deben tener un comentario, antes de su declaración, explicando que hacen.
- Ningún programador debería tener que analizar el código de una función para conocer su utilidad. Tanto el nombre como el comentario que acompañe a la función deben bastar para ello.

Ejemplo:

```
/*
```

```
    Cambiar la configuración de los objetos incluidos en la variable data.
```

```
*/
```

```
function cambiarConfiguraciones ($data) { ... }
```

Funciones:

- Los nombres deben ser verbos o palabras que identifiquen de manera general el objetivo de la función.

Ejemplo:

```
function cambiar datosInternosDelServidor () {...}
```

### 3.4 Pruebas de sistema

El objetivo de las pruebas del software es convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su uso operacional. La prueba es un proceso que intenta proporcionar confianza en el software. Cada vez más el desarrollo del software implica integrar componentes reutilizables y configurar y adaptar software existente para satisfacer requerimientos específicos. Todas las pruebas en tales casos son pruebas del sistema. En un proceso de desarrollo iterativo, las pruebas del sistema se ocupan de probar un incremento que va a ser entregado al cliente; en un proceso en cascada, las pruebas del sistema se ocupan de probar el sistema completo. Para la mayoría de los sistemas complejos, existen dos fases distintas de pruebas del sistema: de integración y de entregas. Las pruebas de integración se ocupan principalmente de encontrar defectos en el sistema. Cuando se descubre un problema, el equipo de integración intenta encontrar la fuente del problema e identificar los componentes que tienen que ser depurados. Las pruebas de entregas son normalmente pruebas de caja negra en las que el equipo de pruebas se ocupa simplemente de demostrar si el sistema funciona o no correctamente. Los problemas son comunicados al equipo de desarrollo cuyo trabajo es depurar el programa. Cuando los clientes se implican en las pruebas de entregas, éstas a menudo se denominan pruebas de aceptación [25].

#### 3.4.1 Pruebas de integración

El proceso de la integración al sistema implica construir este a partir de sus componentes y probar el sistema resultante para encontrar que puedan surgir debido a la integración de los componentes. Las pruebas de integración comprueban que realmente estos componentes funcionan juntos, son llamados correctamente y transfieren los datos correctos en el tiempo preciso a través de sus interfaces [25]. Al



realizar la integración del módulo a Nova NAS para administrar SAV Unix este debe de aparecer en la barra lateral izquierda específicamente en la parte de servicio, puesto que este entra dentro de los servicios a administrar desde Nova NAS (Ver Figura 3). Al realizar las opciones brindadas dentro del módulo para administrar SAV Unix este debe responder correctamente mostrando los mensajes adecuados en cada caso (Ver Figuras 6 y 7).

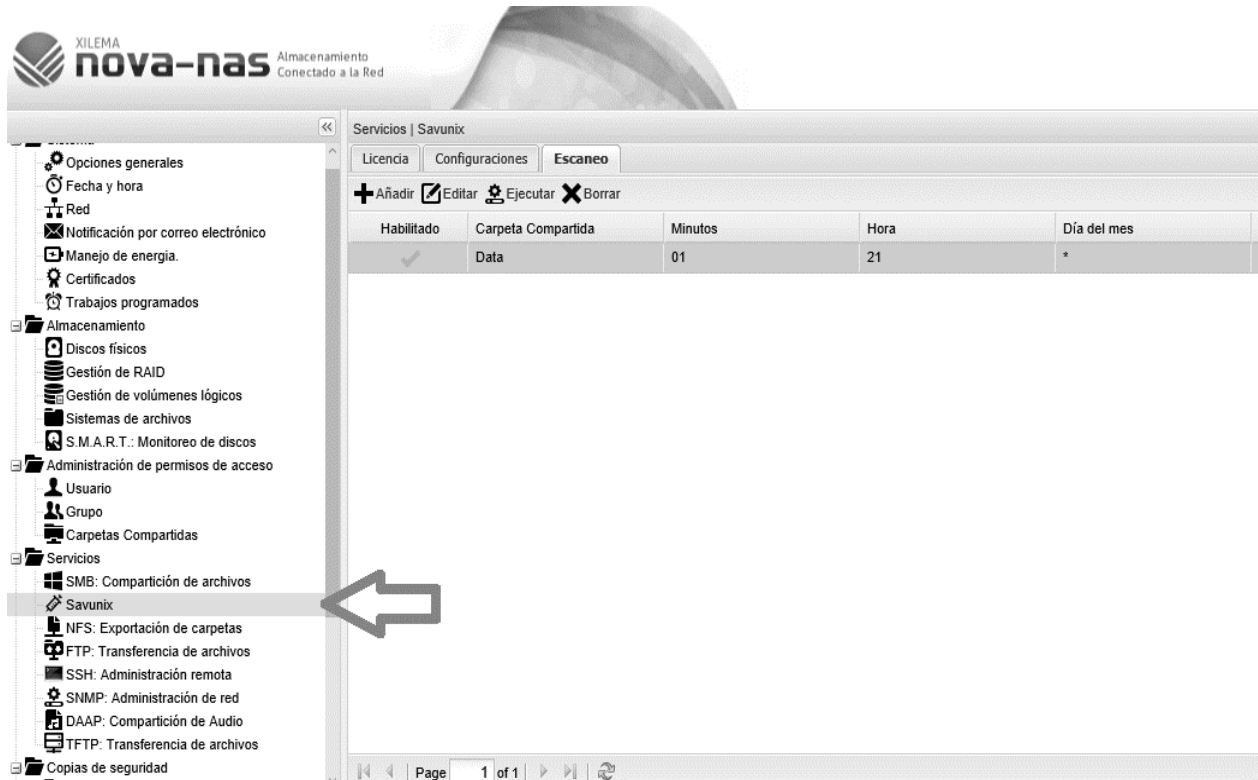


Figura 5: Integración del módulo al sistema.

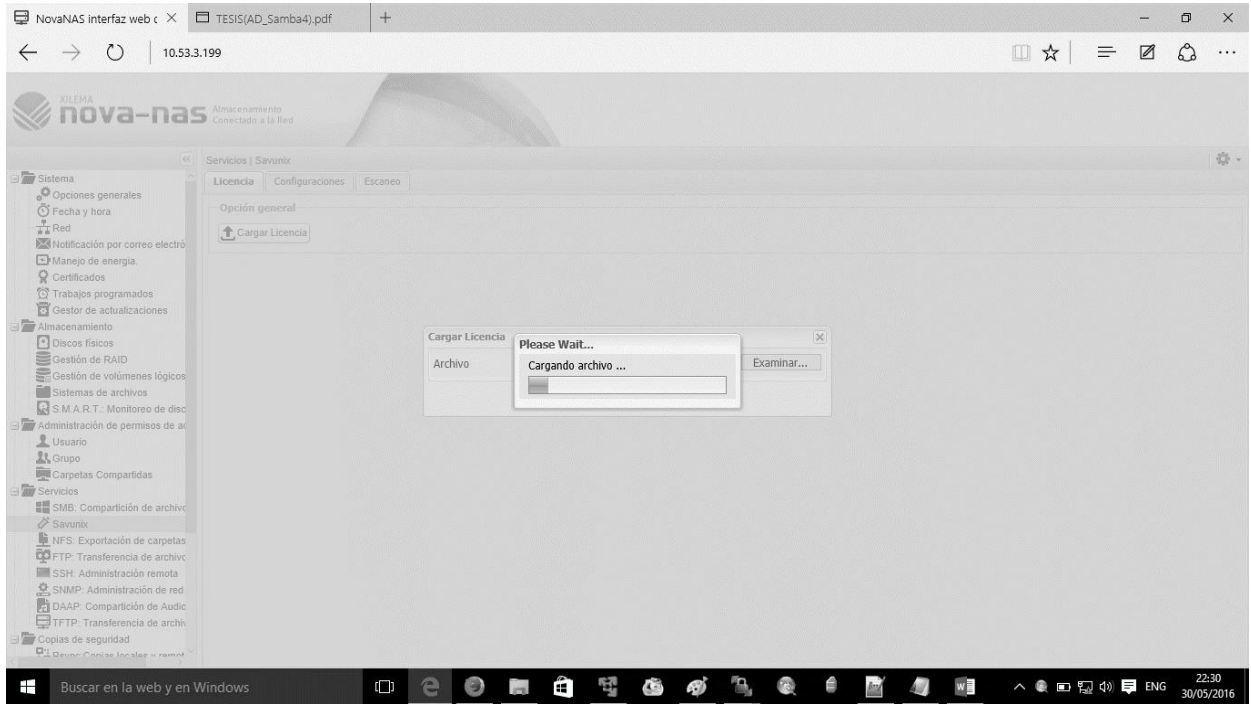


Figura 6: Integración del módulo al sistema\_licencia.

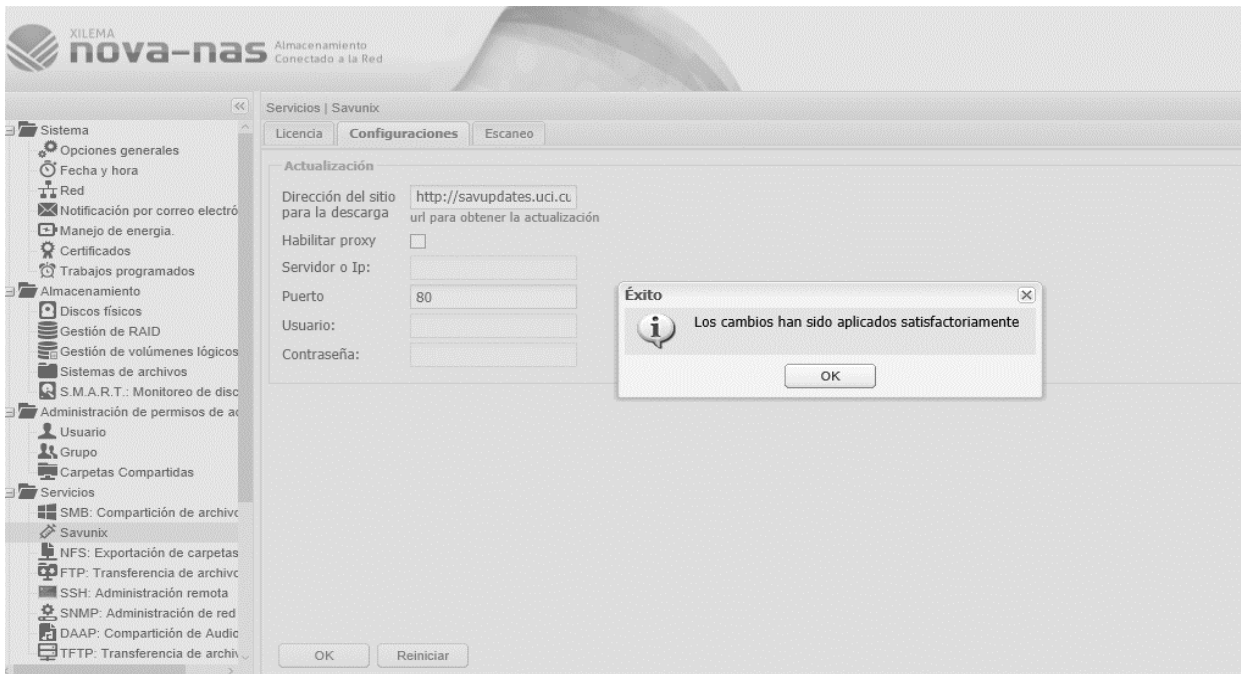


Figura 7: Integración del módulo al sistema\_actualización.

Asumiendo la arquitectura 2-Capas empleada y que el módulo debe incorporarse a una herramienta base, se emplea una estrategia de integración ascendente, donde los componentes se integran de abajo hacia arriba. Las pruebas de integración del módulo para administrar SAV Unix en Nova NAS, demostraron que la integración fue exitosa. Esto permitió que el escaneo de ficheros, se realizara de forma satisfactoria.

### 3.4.2 Pruebas de aceptación

Terminada la implementación se procede a la planeación, diseño y ejecución de las pruebas para garantizar una alta calidad del producto final. Estas no están orientadas a asegurar la ausencia de errores, sino a demostrar que existen defectos en el software, reduciendo así el tiempo transcurrido entre la aparición de un error y su detección. Las pruebas no son un proceso que se realiza una vez desarrollado el software, sino que debe efectuarse en cada una de las etapas de desarrollo. La metodología utilizada propone las pruebas conocidas como pruebas de aceptación, destinadas a evaluar si al final de cada iteración se obtienen los resultados esperados. La planificación de las pruebas se realizó de la siguiente manera:

Tabla 10: Planificación de las pruebas.

No	Tarea	Fecha	Responsable	Participantes	Observaciones
1	Documentar los casos de prueba de aceptación del módulo del sistema Nova NAS.	03/04/2016-10/04/2016	Lázaro Miguel de las Cagigas Barrios	Yasiel Pérez Villazón	Los casos de pruebas se realizan teniendo en cuenta las HU.
2	Realizar las pruebas de aceptación del módulo del sistema Nova NAS.	11/04/2016-18/04/2016	Lázaro Miguel de las Cagigas Barrios	Yasiel Pérez Villazón	Realizar las pruebas al software.

Definidas por el cliente y utilizadas para probar que las HU han sido implementadas correctamente, las pruebas de aceptación tienen como objetivo comprobar que las funcionalidades del sistema se encuentran en relación con las HU definidas. Para las pruebas de aceptación se diseñaron 14 casos de prueba (Ver Anexo 2), de los cuales para una primera iteración se obtuvieron 11 de forma satisfactoria y 3 no satisfactoria asociadas a la falta de validación en los campos del proxy, a los resultados mostrados al realizar el escaneo de un directorio y a la validación de la ruta de actualización. En una segunda iteración se obtuvieron 13 casos de pruebas satisfactorios y solamente 1 no satisfactorio correspondiente al resultado mostrado al realizar el escaneo de un directorio. En la tercera iteración se obtuvieron los 14 casos de pruebas satisfactorios, resultados mostrados en la gráfica siguiente:

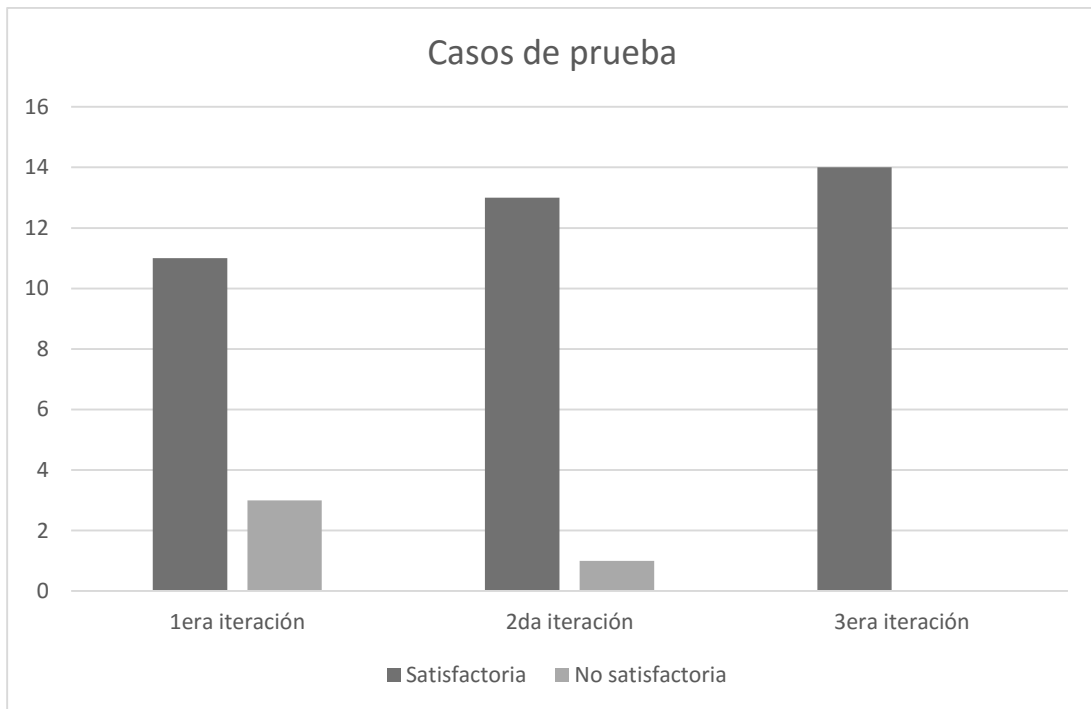


Figura 8: Casos de prueba.

Mediante las pruebas de aceptación, el cliente pudo determinar que se cumple con lo necesario para poner en funcionamiento el módulo para administrar SAV Unix desde Nova NAS, demostrando que las funcionalidades del sistema se encuentran en correspondencia con las HU. Las pruebas de aceptación permitieron evaluar la usabilidad, donde el cliente al interactuar con la aplicación se sintió a gusto y

conforme, destacando altos niveles de usabilidad. Por lo que se concluye que la aplicación funciona correctamente y se encuentra lista para ser usada en entornos reales.

### **3.5 Conclusiones parciales**

La generación de las Tareas de Ingeniería, así como el Plan de Liberación, facilitó el trabajo en el desarrollo de la solución, permitiendo un ahorro considerable de tiempo ya que guiaron el proceso de desarrollo para darle cumplimiento a las Historias de Usuario. El empleo de los estándares de codificación definidos permitió desarrollar un código reutilizable y de fácil comprensión. Una vez desarrolladas las funcionalidades del software, se procedió a la documentación y ejecución de las pruebas de aceptación e integración, con el objetivo de verificar la calidad del producto obteniéndose resultados satisfactorios. A partir de los resultados de estas pruebas se comprobó que el módulo se encuentra listo para ser usado en entornos reales.

## **Conclusiones**

El desarrollo del presente trabajo de diploma permitió elaborar el módulo para la administración del servicio antivirus SAV Unix desde Nova NAS del Centro de Soluciones Libres de la Universidad de las Ciencias Informáticas, dando cumplimiento a los objetivos trazados, destacándose de manera general los siguientes aspectos:

- El análisis de la Resolución No. 127/2007 del MIC permitió determinar a SAV Unix como el más indicado para ser administrado desde Nova NAS.
- El empleo de la arquitectura 2-Capas, así como las herramientas y tecnologías empleadas para el desarrollo de Nova NAS, permitió la reutilización de código y la integración del módulo implementado.
- Los casos de prueba descritos y efectuados arrojaron resultados satisfactorios, siendo la base para conocer que el módulo para administrar SAV Unix, una vez integrado al sistema base, está apto para ser usado en un entorno real de producción.

## Referencias bibliográficas

- [1] Universidad de las Ciencias Informáticas. *Proyecto Estratégico*. s.l. : UCI, 2010.
- [2] Fontela, A. *raiolanetworks.es*. *raiolanetworks.es*. [En línea] [Consultado el: 25 de 5 de 2016.] Disponible en: <https://raiolanetworks.es/blog/desinfectar-virus-de-servidor-linux-con-maldet-y-clamav/>.
- [3] MIC. *RESOLUCION No. 127*. 2007.
- [4] Fumero, G.; Menéndez, Y. J. *Redes, Comunicaciones y el Laboratorio de Informática*.
- [5] Definición de Antivirus - Qué es y Concepto. [En línea] 2006. [Consultado el: 26 de 01 de 2016.] Disponible en: <http://definicion.mx/antivirus/#ixzz32oVtollJ..>
- [6] Características básicas de un antivirus. *CyLDigital.es*. [En línea] [Consultado el: 8 de 2 de 2016.] Disponible en: <http://www.cyldigital.es/articulo/caracteristicas-basicas-de-un-antivirus>.
- [7] Tipos de antivirus informáticos. *tiposde*. [En línea] [Consultado el: 8 de 2 de 2016.] Disponible en: <http://www.tiposde.org/informatica/418-tipos-de-antivirus-informaticos/>.
- [8] SAVUnix. [En línea] [Consultado el: 9 de 2 de 2016.] Disponible en: <http://www.segurmatica.cu/laboratorio/lab2.jsp>.
- [9] Antivirus para Linux File Server | Productos Para Empresas | [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: <http://latam.kaspersky.com/mx/productos-para-empresas/linux-file-server-antivirus#tab1>.
- [10] Baños, R. A. M.; Bonilla M. M. *MÓDULO PARA LA ADMINISTRACIÓN DE SAVUNIX N NOVA PARA SERVIDORES*. La Habana : s.n., 2013.
- [11] Villazón, Y. P. *Administración de un servidor de almacenamiento conectado a la red con Nova NAS*. La Habana : s.n., 2015.
- [12] Definición de Módulo (programación). [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: <http://www.alegsa.com.ar/Dic/modulo.php>.
- [13] Lenguajes de programación. [En línea] [Consultado el: 10 de 2 de 2016.] <http://es.ccm.net/contents/304-lenguajes-de-programacion>.

## Referencias bibliográficas

- [14] PHP: ¿Qué es PHP? - Manual. [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: <http://php.net/manual/es/intro-what-is.php>.
- [15] Definición de PHP. [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: <http://www.alegsa.com.ar/Dic/php.php>.
- [16] El Manual de BASH Scripting Básico para Principiantes/Conceptos e Historia de BASH - Wikilibros. [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: [https://es.wikibooks.org/wiki/El\\_Manual\\_de\\_BASH\\_Scripting\\_B%C3%A1sico\\_para\\_Principiantes/Conceptos\\_e\\_Historia\\_de\\_BASH](https://es.wikibooks.org/wiki/El_Manual_de_BASH_Scripting_B%C3%A1sico_para_Principiantes/Conceptos_e_Historia_de_BASH).
- [17] Sistema UNIX: El shell. [En línea] [Consultado el: 10 de 2 de 2016.] Disponible en: <http://es.ccm.net/contents/692-sistema-unix-el-shell>.
- [18] Soto, E. desarrolloweb.com. *desarrolloweb.com*. [En línea] [Consultado el: 10 de 5 de 2015.] Disponible en: <http://www.desarrolloweb.com/wiki/framework.html#>.
- [19] Design, Develop, and Manage Enterprise Web Applications with Sencha | Sencha. [En línea] [Citado el: 11 de 2 de 2016.] Disponible en: <https://www.sencha.com/>.
- [20] editorbfb . programaciondesarrollo.es. *programaciondesarrollo.es*. [En línea] [Consultado el: 10 de 5 de 2015.] Disponible en: <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.
- [21] Editores de Código . editoresdecodigo.com. *editoresdecodigo.com*. [En línea] [Consultado el: 5 de 10 de 2016.] Disponible en: <http://www.editoresdecodigo.com>.
- [22] Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)). [En línea] [Consultado el: 11 de 2 de 2016.] Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%c3%8d\)%20\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%c3%8d)%20_14720_p/).
- [23] Abad, A. M.; Romero, G. M. P.; García, S. J. *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE*. La Habana : s.n., 2008.
- [24] Reglas de codificación y lineamientos de código PHP, docs.google.com [En línea] [Consultado el: 4 de 6 de 2016.] Disponible en:



## Referencias bibliográficas

[https://docs.google.com/document/d/1rbxDfM0zsbFDNRZeM2FoXfRDbyYSiSt6tCdbYPA0qdzs/edit?hl=en\\_US&pref=2&pli=1](https://docs.google.com/document/d/1rbxDfM0zsbFDNRZeM2FoXfRDbyYSiSt6tCdbYPA0qdzs/edit?hl=en_US&pref=2&pli=1).

[25] SOMMERVILLE, I. *Ingeniería del software Séptima edición* . s.l. : Pearson Educación, 2005.

## ANEXOS

### ANEXO 1

#### Entrevista a Ing. Juan Carlos Casas, especialista del Grupo Soporte Técnico de la Empresa Segurmática, empresa especializada en consultoría y seguridad informática.

Objetivo: Investigar sobre los antivirus de producción nacional u otro autorizado oficialmente para su uso en el país.

#### CUESTIONARIO

1. ¿Qué papel desempeña Segurmática en la sociedad cubana?
2. ¿Qué antivirus son desarrollados en Cuba?
3. ¿De estos antivirus nacionales cuáles son funcionales en Linux?
4. ¿Qué antivirus internacionales pueden ser utilizados en Cuba y por qué?
5. ¿Puede en un futuro cambiar la situación en relación a los antivirus internacionales a utilizar en Cuba?
6. ¿Qué aspectos considera que distinguen a los antivirus nacionales de los internacionales autorizados?

### ANEXO 2

#### Casos de prueba.

Escenarios vinculados a la HU Adicionar licencia:

Tabla 11: Caso de prueba ruta correcta para la licencia.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU2	<b>Nombre Historia de Usuario:</b> Adicionar licencia.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar añadir licencia al servicio antivirus SAV Unix.	

<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.
<b>Entrada / Pasos de ejecución:</b> 1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios 2. Seleccionar <b>Licencia</b> 3. Introducir la ruta de fichero con extensión .lic y fecha de uso correcta 4. Dar clic en el botón OK
<b>Resultado Esperado:</b> El servicio antivirus se activa.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 12: Caso de prueba ruta incorrecta para la licencia.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU2_1	<b>Nombre Historia de Usuario:</b> Adicionar licencia.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que valida la ruta de licencia para el servicio antivirus SAV Unix.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> 1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios 2. Seleccionar <b>Licencia</b> 3. Introducir la ruta del fichero sin extensión .lic 4. Dar clic en el botón OK	
<b>Resultado Esperado:</b> Muestra el mensaje de “Archivo incorrecto”.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 13: Caso de prueba ruta con fecha incorrecta para la licencia.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU2	<b>Nombre Historia de Usuario:</b> Adicionar licencia.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Añadir licencia al servicio antivirus SAV Unix con fecha de licencia caduca.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Licencia</b></li> <li>3. Introducir la ruta de un fichero .lic con fecha caduca</li> <li>4. Dar clic en el botón OK</li> </ol>	
<b>Resultado Esperado:</b> El servicio antivirus muestra mensaje de error: "Fecha caduca".	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 14: Caso de prueba ruta vacía para la licencia.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU2_1	<b>Nombre Historia de Usuario:</b> Adicionar licencia.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que la ruta de licencia no puede ser vacía, para configurar servicio antivirus SAV Unix.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	

<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Licencia</b></li> <li>3. Dejar campo vacío</li> <li>4. Dar clic en el botón OK</li> </ol>
<p><b>Resultado Esperado:</b></p> <p>Muestra un mensaje de requerimiento de fichero.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>

Escenarios vinculados a la HU Configurar actualización:

Tabla 15: Caso de prueba ruta correcta para actualizar.

<b>Caso de Prueba de Aceptación</b>	
<p><b>Código Caso de Prueba:</b></p> <p>NovaNAS- Antivirus_HU3</p>	<p><b>Nombre Historia de Usuario:</b></p> <p>Configurar actualización.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón</p>	
<p><b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix con la ruta correcta se actualiza.</p>	
<p><b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.</p>	
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Introducir ruta correcta para la actualización</li> <li>4. Dar clic en el botón actualizar</li> </ol>	
<p><b>Resultado Esperado:</b></p> <p>El servicio antivirus se actualiza.</p>	
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>	

Tabla 16: Caso de prueba ruta incorrecta para actualizar.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU3_1	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix con la ruta incorrecta no se actualiza.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Introducir la ruta de la actualización incorrecta</li> <li>4. Dar clic en el botón actualizar</li> </ol>	
<b>Resultado Esperado:</b> Muestra el mensaje de: "Error en la descarga".	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 17: Caso de prueba campo Servidor o Ip vacío en el proxy.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU3_2	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix valida el proxy para el campo Servidor o Ip.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	

<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Dejar el campo Servidor o Ip vacío en el proxy</li> <li>4. Dar clic en el botón actualizar</li> </ol>
<p><b>Resultado Esperado:</b></p> <p>Muestra un mensaje de requerimiento de campo Servidor o Ip.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>

Tabla 18: Caso de prueba campo Puerto vacío en el proxy.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b>	<b>Nombre Historia de Usuario:</b>
NovaNAS- Antivirus_HU3_2	Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix valida el proxy para el campo Puerto.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Dejar el campo Puerto vacío en el proxy</li> <li>4. Dar clic en el botón actualizar</li> </ol>	
<p><b>Resultado Esperado:</b></p> <p>Muestra un mensaje de requerimiento de campo Puerto.</p>	
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>	

Tabla 19: Caso de prueba campo Usuario vacío en el proxy.

<b>Caso de Prueba de Aceptación</b>
-------------------------------------

<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU3_2	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix valida el proxy para el campo Usuario.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Dejar el campo Usuario vacío en el proxy</li> <li>4. Dar clic en el botón actualizar</li> </ol>	
<b>Resultado Esperado:</b> Muestra un mensaje de requerimiento de campo Usuario.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 20: Caso de prueba campo Contraseña en el proxy.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU3_2	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix valida el proxy para el campo Contraseña.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Dejar el campo Contraseña en el proxy</li> <li>4. Dar clic en el botón actualizar</li> </ol>	



<b>Resultado Esperado:</b> Muestra un mensaje de requerimiento de campo Contraseña.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 21: Caso de prueba valores incorrectos en el proxy.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU3_2	<b>Nombre Historia de Usuario:</b> Configurar actualización.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar que el servicio antivirus SAV Unix valida el proxy cuando se le introducen valores incorrectos en cualquiera de sus campos.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Configuraciones</b></li> <li>3. Establecer valores incorrectos en el proxy</li> <li>4. Dar clic en el botón actualizar</li> </ol>	
<b>Resultado Esperado:</b> Muestra mensaje: "Error en el Proxy".	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Escenario vinculado a la HU Escanear directorio:

Tabla 22: Caso de prueba escanear directorio.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU4	<b>Nombre Historia de Usuario:</b> Escanear directorio.

<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón
<b>Descripción de la Prueba:</b> Probar el escaneo de un directorio.
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.
<b>Entrada / Pasos de ejecución:</b> 1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios 2. Seleccionar <b>Escaneo</b> 3. Agregar directorio a escanear 4. Se selecciona Ejecutar 5. Se selecciona Comienzo
<b>Resultado Esperado:</b> Se escanea un directorio mostrando los datos relevantes del mismo de forma organizada (ficheros revisados, ficheros infectados, ficheros sospechosos y ficheros descontaminados) y se notifica por correo al administrador.
<b>Evaluación de la Prueba:</b> Satisfactoria

Tabla 23: Caso de prueba escaneo programado de directorio.

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU4	<b>Nombre Historia de Usuario:</b> Escanear directorio.
<b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón	
<b>Descripción de la Prueba:</b> Probar el escaneo programado de un directorio.	
<b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.	
<b>Entrada / Pasos de ejecución:</b> 1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios 2. Seleccionar <b>Escaneo</b> 3. Agregar directorio a escanear, establecer un tiempo para el mismo y activar	

<p><b>Resultado Esperado:</b> Pasado el tiempo establecido se escanea el directorio.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>

Tabla 24: Caso de prueba escaneo programado de directorio no activado.

<b>Caso de Prueba de Aceptación</b>	
<p><b>Código Caso de Prueba:</b> NovaNAS- Antivirus_HU4</p>	<p><b>Nombre Historia de Usuario:</b> Escanear directorio.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Yasiel Pérez Villazón</p>	
<p><b>Descripción de la Prueba:</b> Probar el escaneo programado de un directorio no activado.</p>	
<p><b>Condiciones de Ejecución:</b> El servicio antivirus SAV Unix debe estar instalado.</p>	
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar <b>SAV Unix</b> en la parte de administración de servicios</li> <li>2. Seleccionar <b>Escaneo</b></li> <li>3. Agregar directorio a escanear, establecer un tiempo para el mismo y se deja sin activar</li> </ol>	
<p><b>Resultado Esperado:</b> Pasado el tiempo establecido no se escanea el directorio.</p>	
<p><b>Evaluación de la Prueba:</b> Satisfactoria</p>	