

# **Universidad de las Ciencias Informáticas**

## **Facultad 1**




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

**Título: “Módulo para administrar el servidor web Nginx desde la Herramienta para la Migración y Administración de Servicios Telemáticos”**

**Autora:** Rachel Molina Rodríguez

**Tutores:** Ing. Nurisel Palma Pérez  
Ing. Delvis Mesa Chaviano

**La Habana, junio 2017**  
**“Año 59 de la Revolución”**



*"Estoy convencido que la mitad de lo que separa a los emprendedores exitosos de los que han fracasado, es la perseverancia."*

*Steve Jobs*

## **Declaración de autoría**

Declaro por este medio que yo Rachel Molina Rodríguez, con carnet de identidad 94070134013 soy autora principal del trabajo titulado “Módulo para administrar el servidor web Nginx desde la Herramienta para la Migración y Administración de Servicios Telemáticos” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales, con carácter exclusivo.

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

Rachel Molina Rodríguez

\_\_\_\_\_

Firma de Autor

Ing. Nurisel Palma Pérez

\_\_\_\_\_

Firma de Tutor

Ing. Delvis Mesa Chaviano

\_\_\_\_\_

Firma de Tutor

**Tutores:**

Ing. Inst. Nurisel Palma Pérez, graduada de Ingeniera en Ciencias Informáticas en el 2013 en la Universidad de las Ciencias Informáticas (UCI). Posee la categoría docente de Instructor. Forma parte del departamento Servicios Integrales en Migración, Asesoría y Sistemas (SIMAYS) donde se desempeña como desarrolladora de la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST). Posee publicaciones y participación en eventos como el II Congreso Internacional de Ingeniería Informática y Sistemas de Información, la Conferencia Científica de la UCI en sus dos ediciones y el III Congreso Multidisciplinario de Ciencias Aplicadas en Latinoamérica. (npalma@uci.cu)

Ing. Delvis Mesa Chaviano, graduado de Ingeniero en Ciencias Informáticas en el 2015 en la Universidad de las Ciencias Informáticas (UCI). Forma parte del departamento Servicios Integrales en Migración, Asesoría y Sistemas (SIMAYS) donde se desempeña como desarrollador del proyecto Nova Servidores. (dmesa@uci.cu)

*A Dios por permitirme llegar a este momento de mi vida.*

*A mi mamá por ser la persona que me trajo al mundo, por dedicarse en cuerpo y alma a sus hijas, por hacer posible mis días en esta universidad sin tener que preocuparme por algo más que no fuera estudiar, por ser mi motivo esencial para llegar a ser ingeniera, pero sobre todo por ser mi mamá, no será la mejor del mundo, pero para mí lo es, es mi guerrera favorita y mi razón de ser. ¡Gracias mami!!!*

*A mi niña hermosa, por preocuparse por mí, por estar siempre que la necesito, por secar mis lágrimas en días difíciles, aunque sea por teléfono, por apoyarme y darme fuerzas hasta en los momentos que ni siquiera yo creí que podía lograrlo, por confiar en mí y por seguir mis pasos porque en algún momento tienes que ser ingeniera. ¡Eso es una orden de tu hermana mayor!!!*

*A mi papá por ser mi guía y mi espíritu para lograr superarme, por dejarme seguir sus pasos, por su apoyo incondicional para que este día fuera el mejor, por exigirme lo suficiente para cumplir mis metas, por estar pendiente de mí y de mi tesis, por estar ahí incluso para revisar mi documento sin apenas entenderlo completamente, por su preocupación como padre y como profesional. Gracias papi.*

*A mi gordo, por ser más que un padrastro en mi vida y en mis años de universidad, por ser mi segundo papá, por estar ahí en el momento en que lo necesito sin tener que decírselo, por la confianza, el apoyo, los regaños si hubo alguno, la preocupación y el interés porque saliera adelante. No sé que me querías doctora, pero ni modo, me convertí en ingeniera.*

*A mis tutores por ayudarme en todo lo que pudieron, especialmente a Nuri (la mejor tutora del Ejército Libertador), por los regaños, por estar pendiente de mí para que mi tesis saliera adelante, por sentarse conmigo siempre que la necesité, por confiar en mí y darme su apoyo cuando todo parecía imposible, por cada minuto que dedicó a mi tesis a pesar de su trabajo, de sus problemas, eso nunca impidió que sus tesis salieran ilesos en una súper revisión de tesis, de verdad gracias por todo.*

*A Yosel y Haniel, por ser mis tutores porque yo lo decidí, por adoptarme en sus vidas profesionales, gracias por la ayuda en todo momento, por no mandarme bien lejos siempre que los molesté, por los cafés juntos para poder seguir trabajando y por cada minuto que dejaron de hacer lo suyo para atenderme a mí.*

*A mis viejitas Loyda y Aleida, por ser las mejores abuelas que se puede tener y por malcriarme todavía como a una niña.*

*A mis hermanos Osky, la pequeña, Zula y a Trielis porque a pesar de la distancia también se preocuparon por mí.*

*A mi familia por el apoyo y la preocupación, mi tía Loyda que estuvo ahí en todo momento de mi carrera que la necesité, todos mis tíos, mis primos, Ale más que mi primo mi hermano. Gracias a todos.*

*A Zuleica, que ha sabido exigirme y preocuparse por mí tanto como mi mamá y aconsejarme como a su propia hija.*

*A mis niñas Stephany y Zuleymis por ser mis mejores amigas, mis hermanas y mi mamá cuando tenían que serlo, gracias por los momentos compartidos, los buenos, los malos y por los que vendrán.*

*A mi amiga Clau que, aunque esté lejos, sé que hoy va a estar muy orgullosa de mí.*

*A mis hijos Javier Hosford y César Javier y a los que me abandonaron por otra madre, ellos saben quiénes son, gracias por los momentos compartidos y por desestresarme de la tesis, porque lo de ellos son las fiestas. Gracias de veras mis niños.*

*A mis mejores amigos: Javier Piñeiro, Maikel, Osmel, Leonard, Robert, Yosel una vez más y Alien (mi padre) por estar ahí y correr conmigo siempre que lo necesitara, casi siempre por el hambre o los dolores de ovario.*

*A mis compañeros de apto y del grupo durante estos 5 años por los momentos vividos, los felices y los estresantes.*

*A Lázaro porque a pesar de no estar juntos compartimos todos estos años como amigos, como novios, como ex y como amigos una vez más. Gracias por los consejos, por tu apoyo siempre que lo necesité y por hacer que mi paso por esta universidad valiera la pena.*

*A todas mis amistades que están siempre ahí, estemos cerca o lejos, gracias a todos.*

*A las dos personas más importantes de mi vida:*

*A mi mami que es la razón de mis fuerzas para seguir adelante.*

*A mi mejor amiga, mi princesa y la niña de mis ojos, mi hermana Ana Laura porque  
quiero que en algún momento sea ella quien esté delante dedicándome su trabajo de  
diploma.*



**Resumen**

El presente trabajo se centró en el objetivo de desarrollar un módulo que permita la administración del servidor web Nginx en la Herramienta para la Migración y Administración de Servicios Telemáticos. Se emplearon los métodos Histórico-Lógico, Analítico-Sintético, Modelación y la Entrevista como técnica de recopilación de información. Se documentaron las diferentes tecnologías a usar y la metodología de desarrollo de software Proceso Unificado Ágil en su versión para la Universidad de las Ciencias Informáticas, como guía del proceso. Finalmente se presentó un módulo que permite la administración de Nginx en diferentes computadoras servidoras de forma remota. El módulo posee las funcionalidades necesarias para configurar las conexiones en el servidor, gestionar los hosts virtuales y especificar parámetros para el tráfico mediante el Protocolo de Transferencia de Hipertexto. Con el resultado obtenido la herramienta base se puede desplegar y adaptar fácilmente a las empresas cubanas que publican su contenido con Nginx.

**Palabras Claves:** administración, Nginx, remota, servidor.

## Índice

Introducción.....	1
<b>Capítulo 1: Fundamentación teórica .....</b>	<b>6</b>
1.1 Servidor web.....	6
1.2 Nginx .....	7
1.2.1 Características de Nginx.....	7
1.2.2 Configuración de Nginx .....	9
1.3 Descripción de HMAST.....	9
1.3.1 Arquitectura de HMAST .....	10
1.3.2 Funcionalidades que ofrece HMAST .....	11
1.3.3 Consideraciones para implementar un módulo para HMAST .....	12
1.4 Herramientas que administran Nginx .....	12
1.5 Herramientas y tecnologías .....	14
1.6 Metodología de desarrollo de software .....	18
Conclusiones parciales .....	18
<b>Capítulo 2: Análisis y diseño del módulo Nginx .....</b>	<b>20</b>
2.1 Propuesta de solución.....	20
2.2 Artefactos generados .....	20
2.2.1 Especificación de requisitos .....	21
2.2.2 Historias de Usuario .....	24
2.3 Arquitectura del módulo .....	30
2.4 Diagrama de paquetes.....	31
2.4.1 Diagrama de clases de las entidades.....	33
2.5 Patrones de diseño.....	33
2.5.1 Patrones GRASP.....	34
2.5.2 Patrones GoF.....	34
Conclusiones parciales .....	35
<b>Capítulo 3: Implementación y pruebas al módulo Nginx.....</b>	<b>36</b>
3.1 Estándar de codificación .....	36

<b>3.2 Estrategia de pruebas</b> .....	37
<b>3.2.1 Pruebas internas</b> .....	38
<b>3.2.2 Pruebas de aceptación</b> .....	43
<b>Conclusiones parciales</b> .....	44
<b>Conclusiones generales</b> .....	45
<b>Recomendaciones</b> .....	46
<b>Referencias Bibliográficas</b> .....	47
<b>Anexo 1: Entrevista al administrador del Nodo Central en la UCI.</b> .....	49
<b>Anexo 2: Historias de Usuario.</b> .....	49
<b>Anexo 3: Acta de aceptación del producto.</b> .....	70

## Índice de tablas

Tabla 1. Listado de requisitos funcionales.....	21
Tabla 2: Listado de requisitos no funcionales (elaboración propia).....	23
Tabla 3: HU Instalar el servidor web Nginx (elaboración propia).....	24
Tabla 4: HU Adicionar host virtual.....	25
Tabla 5: HU Listar host virtual. ....	27
Tabla 6: HU Editar conexiones.....	28
Tabla 7: HU Eliminar host virtual. ....	29
Tabla 8: Código de la funcionalidad updateConnections . ....	39
Tabla 9: Listado de caminos independientes. ....	41
Tabla 10: Caso de prueba de unidad para el camino 1. ....	41
Tabla 11: Caso de prueba de unidad para el camino 2. ....	42
Tabla 12: Caso de prueba de unidad para el camino 3. ....	42
Tabla 13: HU Desinstalar el servidor web Nginx.....	49
Tabla 14: HU Mostrar host virtual. ....	50
Tabla 15: HU Editar host virtual.....	51
Tabla 16: HU Iniciar el servidor web Nginx.....	54
Tabla 17: HU Detener el servidor web Nginx. ....	54
Tabla 18: HU Reiniciar el servidor web Nginx.....	55
Tabla 19: HU Recargar el servidor web Nginx. ....	55
Tabla 20: HU Habilitar host virtual. ....	56
Tabla 21: HU Deshabilitar host virtual.....	57
Tabla 22: HU Mostrar Conexiones.....	58
Tabla 23: HU Mostrar las configuraciones del tráfico HTTP.....	59

Tabla 24: HU Editar las configuraciones del tráfico HTTP. ....	60
Tabla 25: HU Listar directorio.....	61
Tabla 26: HU Adicionar directorio. ....	62
Tabla 27: HU Editar directorio. ....	63
Tabla 28: HU Eliminar directorio. ....	65
Tabla 29: HU Mostrar directorio. ....	66
Tabla 30: HU Aplicar cambios locales.....	67
Tabla 31: HU Descartar cambios locales. ....	68

## Introducción

Internet tiene un impacto profundo en el mundo laboral, el ocio y el conocimiento a nivel mundial. Uno de los servicios que más éxito ha tenido en internet ha sido la *World Wide Web* (WWW o la web). Con la web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea. Este nuevo medio de comunicación logró romper las barreras físicas entre regiones. La mayoría de las industrias de comunicación están siendo transformadas o redefinidas por el internet, permitiendo el nacimiento de nuevos servicios. Existe una gran tendencia hacia la creación de aplicaciones web debiendo la popularidad a la portabilidad del navegador web y su independencia del sistema operativo, así como a la facilidad para actualizar y mantener las mismas sin distribuir e instalar software a miles de usuarios potenciales (Camilo, 2012).

En la actualidad los usuarios acceden a una aplicación web a través de internet, mediante un navegador o cliente utilizando el Protocolo de Transferencia de Hipertexto, (HTTP, del inglés *Hypertext Transfer Protocol*) para la comunicación con el servidor web. Un servidor web o servidor HTTP es un programa informático que procesa cualquier aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con un cliente (Bautista, 2011). Un servidor web tiene asignada la responsabilidad de publicar contenido en forma de hipertextos, páginas web o páginas elaboradas con el Lenguaje de Marcado para Hipertexto, (HTML del inglés *HyperText Markup Language*), que son almacenados en hosts virtuales<sup>1</sup> y accedidos por los usuarios. Según las estadísticas de la compañía inglesa Netcraft<sup>2</sup> en marzo del 2017, los dos servidores web de código abierto más usados son Apache 2 y Nginx, con 22% y 20% de uso respectivamente (Netcraft, 2017).

Apache 2 se emplea en más de 380 millones de sitios a nivel mundial (Netcraft, 2017). En las empresas cubanas una de las vías para su administración es mediante el módulo ofrecido por la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST), desarrollada en el Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI). El desarrollo de HMAST tiene el propósito de disponer de una aplicación que se emplee en las diferentes empresas e instituciones

---

<sup>1</sup> Hosts virtuales: se refiere a que en un mismo servidor web se pueden hospedar múltiples proyectos cada uno con su propio dominio, aunque todos pertenezcan a la misma dirección IP (del inglés *Internet Protocol*) pública.

<sup>2</sup> Netcraft: es una compañía de servicios de internet que ofrece análisis de cuota de mercado de servidores y alojamiento web.

cubanas que hagan uso, en la mayoría de las PC (del inglés *Personal Computer*) servidoras<sup>3</sup>, de sistemas operativos privativos (Palma, 2013). HMAST es una aplicación web que permite administrar y migrar los servicios telemáticos a plataformas libres. Cuenta con un sistema base que permite la administración de las PC servidoras de forma remota, el cual posee funcionalidades necesarias para administrar los usuarios y los servicios. La versión 2.0 cuenta con cinco módulos: DHCP, MySQL, Bacula, OpenSSH y Apache2.

Por otra parte, Nginx se emplea en más de 350 millones de sitios a nivel mundial. Internacionalmente en grandes empresas de internet como: Tumblr, WordPress, Instagram, Yahoo, YouTube, Pinterest, Zynga, SourceForge, GitHub, DropBox, Intel, NetFlix, Facebook en el servidor de descarga de archivos zip pesados, entre otras compañías, por ejemplo: algunas Redes de Entrega de Contenidos (CDN, del inglés, *Content Delivery Network*), Cloudflare, NetDNA/MaxCDN, Cachefly, ProCDN/MediaTemple (Netcraft, 2017).

En Cuba es utilizado en sitios web por distintas empresas y servicios como son: el grupo hotelero Islazul, Radio Cadena Habana, el Sistema de las Naciones Unidas, el sitio oficial de promoción de turismo cubano, Las Terrazas, Radio Musical de Cuba, el periódico Granma, entre otros. En la UCI importantes sitios como InterNos, Firefoxmanía, Migración UCI, Sistema de Gestión Universitaria, Periódico Mella, Plataforma Educativa, Revista Cubana de Ciencias Informáticas, publican su contenido con Nginx.

Características como el ser ligero, multiplataforma y fácil de instalar, además de que puede ser utilizado junto a Apache, convierten a Nginx en una alternativa viable para muchas empresas en Cuba y en el mundo. En muchas de ellas solo usan Nginx para servir contenido estático y Apache para el contenido dinámico, aunque en muchas otras sobre Nginx recae la principal carga de la administración de los servicios.

La adaptabilidad de HMAST al entorno empresarial cubano, depende en gran medida de contar con los módulos necesarios para administrar y configurar los más disímiles servicios en las empresas cubanas. En la actualidad HMAST no permite la gestión de las configuraciones de Nginx, limitando el despliegue de la herramienta en ecosistemas en los cuales se emplee este servidor. La versatilidad y potencia de Nginx a partir del gran número de directivas específicas que posee y las configuraciones de las mismas, hacen

---

<sup>3</sup> PC servidoras: computadora con un software servidor.

que su administración pueda ser compleja para el marco de la Pequeña y Mediana Empresa Cubana (PYMES).

Tomando como punto de partida la situación descrita, la presente investigación se plantea como **problema a resolver**: ¿Cómo realizar la administración del servidor web Nginx desde HMAST?

El problema planteado anteriormente tiene como **objeto de estudio** el servidor web Nginx y el **campo de acción** estará enmarcado en la administración del servidor web Nginx.

El **objetivo general** de este trabajo es desarrollar un módulo que permita la administración del servidor web Nginx en HMAST.

Para un mejor desarrollo de la investigación se plantean los siguientes **objetivos específicos**:

1. Caracterizar la administración del servidor web Nginx para sentar las bases teóricas de la investigación.
2. Analizar un módulo para HMAST que permita la futura implementación de la solución.
3. Diseñar un módulo para HMAST que permita la futura implementación de la solución.
4. Implementar la solución propuesta para darle cumplimiento al objetivo de la investigación.
5. Realizar las pruebas correspondientes al módulo implementado.

Para guiar el desarrollo del trabajo se plantean las siguientes **preguntas científicas**:

1. ¿Cuáles son los presupuestos teóricos que fundamentan la administración de servicios para el servidor web Nginx?
2. ¿Cuáles son las herramientas y tecnologías más adecuadas para diseñar un módulo que permita la administración del servidor web Nginx para HMAST?
3. ¿Qué métodos o técnicas aplicar para verificar la calidad del módulo de administración del servidor web Nginx para HMAST?

Las **tareas de la investigación** planteadas son:

- Caracterización de la administración de Nginx.
- Definición de los requisitos funcionales y no funcionales a implementar.

- Especificación de las Historias de Usuario.
- Descripción de la arquitectura del módulo.
- Implementación del módulo definido.
- Diseño de casos de prueba.
- Realización de pruebas al módulo implementado.

Para el cumplimiento del objetivo propuesto y de las tareas de investigación se emplean métodos teóricos y empíricos de la investigación científica. Los **métodos teóricos** que se utilizan son:

**Histórico-Lógico** en el estudio de las tesis de pregrado e investigaciones realizadas en torno a HMAST para identificar las principales características de la misma, así como las soluciones dadas a problemas similares a los de la presente investigación.

**Analítico-Sintético** se pone en práctica para el estudio de diferentes fuentes bibliográficas y de los elementos del problema planteado, profundizando en las características del servidor web Nginx, para luego sintetizarlos y elaborar la propuesta de solución de la investigación.

**Modelación**, para la creación de abstracciones como los diagramas, con el objetivo de explicar la realidad. En el marco del diseño del módulo propuesto se modeló el diagrama de paquetes.

Como técnica de recopilación de información se utiliza la **Entrevista** para investigar sobre el funcionamiento y las principales configuraciones de Nginx, así como la existencia de alguna herramienta en la UCI que realice la administración de dicho servidor.

El presente documento está estructurado en introducción, tres capítulos, conclusiones generales, referencias bibliográficas, anexos y glosario de términos.

- En el primer capítulo “Fundamentación teórica” se realiza el estudio del estado del arte donde se hace referencia a los principales conceptos utilizados en la presente investigación. Se realiza un análisis bibliográfico basándose en los métodos teóricos expuestos anteriormente, que permite caracterizar el servidor web Nginx. Además, se describe HMAST, su arquitectura, así como la metodología de desarrollo a emplear. Se definen las herramientas y tecnologías que se utilizarán para el desarrollo del módulo.



- En el segundo capítulo “Análisis y diseño del módulo Nginx” se exponen las características principales del módulo y se especifican las funcionalidades a implementar. Se define la arquitectura del sistema y se representan los patrones de diseño que se utilizarán en la implementación, además se presentan las Historias de Usuario correspondientes a las funcionalidades y los prototipos de interfaz de usuario.
- En el tercer capítulo “Implementación y pruebas al módulo Nginx” se realiza la implementación de la propuesta de solución y se describen, diseñan, realizan y controlan los casos de prueba que se aplicarán sobre el sistema.

### **Capítulo 1: Fundamentación teórica**

En el presente capítulo se realiza un estudio de los servidores web, centrándose en el servidor web Nginx. Se describe HMAST, así como su arquitectura y se exponen las consideraciones a tener en cuenta para la integración del módulo a esta herramienta. Además, se detallan las herramientas y tecnologías a utilizar para el desarrollo del módulo.

#### **1.1 Servidor web**

Un servidor web es un programa informático que procesa cualquier aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con un cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente (Romero, 2012). Un servidor web se mantiene a la espera de peticiones de ejecución que le hará un cliente o un usuario de internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

Existen varios tipos de servidores web, entre los que se encuentran:

- **Servidores basados en procesos:** se basan en la obtención de paralelismo mediante la duplicación del proceso de ejecución. Es el predecesor de los demás diseños. El más simple de este tipo de diseños es en el que el proceso principal con la llegada de una nueva conexión se duplica creando una copia exacta que atenderá la misma (Palma, 2013).
- **Servidores basados en hilos:** le son aplicables los conceptos básicos respecto al funcionamiento de un servidor basado en procesos, heredando muchas de sus características, entre ellas la de la simplicidad en su diseño e implementación. Las principales diferencias de los dos modelos residen en el propio concepto de hilo. Tienen la ventaja que la creación de un hilo es menos costosa que la de un proceso. Varios hilos de un mismo proceso comparten el mismo espacio de memoria, propiciando que puedan compartir datos entre ellos, sin embargo, esto implica un riesgo de seguridad (Palma, 2013).

- **Servidores basados en sockets no bloqueantes o dirigidos por eventos:** que basan su funcionamiento en la utilización de lecturas y escrituras asíncronas sobre sockets. Utilizan una llamada al sistema que examine el estado de los sockets operativos que implementan una o más funciones de examen de sockets. Estas funciones tienen como objetivo inspeccionar el estado de un grupo de sockets asociados a cada una de las conexiones. Este tipo de servidores a pesar de caracterizarse por su velocidad, tienen la desventaja que la concurrencia es simulada, existiendo un solo proceso y un solo hilo, desde el cual se atienden todas las conexiones (Palma, 2013).

Nginx es un servidor web basado en sockets no bloqueantes o dirigido por eventos. En el siguiente epígrafe se hace referencia al funcionamiento, características y configuraciones de dicho servidor.

### 1.2 Nginx

Nginx es un servidor web HTTP de código abierto licenciado bajo la Licencia de Distribución de Software de Berkeley (BSD, del inglés *Berkeley Software Distribution*) simplificada, un servidor proxy de correo, y un servidor genérico TCP/UDP<sup>4</sup> proxy, creado como un proyecto personal de un administrador de sistemas ruso llamado Ígor Sysóiev, quien comenzó a trabajar en el proyecto en 2002, y el primer código público salió en octubre del mismo año. Al igual que muchos líderes de proyectos de código abierto, estaba tratando de satisfacer un deseo y al mismo tiempo, trabajaba para Rambler, un portal de internet ruso de rápido crecimiento y necesitaba un servidor que pudiera manejar más tráfico que las alternativas de código abierto. En Nginx, todo se ejecuta desde el mismo proceso y su funcionamiento está basado en eventos, quedando todo más centralizado y ahorrando una gran cantidad de recursos en contenido estático.

#### 1.2.1 Características de Nginx

El servidor web Nginx se caracteriza por:

- El soporte nativo para caché de datos y la posibilidad de configurar un balanceo de cargas entre los diferentes servidores que se tienen contratados.

---

<sup>4</sup> TCP/UDP: protocolos de la capa de transporte.

- Es un servidor web/proxy multiplataforma, inicialmente creado para funcionar en sistemas operativos Unix, más tarde apareció una versión compatible con Windows, y quedó listo para trabajar en Linux, Unix, Windows, MAC IOS, Solaris y BSD.
- Servidor de ficheros estáticos: los módulos no pueden ser cargados de forma dinámica, deben ser incluidos en la compilación. Además, no se pueden desactivar en tiempo de ejecución debido a que están compilados e integrados en el binario principal.
- Arquitectura orientada a eventos para el manejo de peticiones: en lugar de abrir subprocessos para cada petición, Nginx utiliza una arquitectura basada en eventos que hace que su consumo de recursos crezca de forma predecible. En el momento que se inicia el servidor, existe en memoria solo el proceso principal. Este no es el que procesa las peticiones hechas por el cliente, sino otros procesos generados por él. Desde el archivo de configuración se puede definir la cantidad de procesos, las conexiones máximas por proceso y otros parámetros.
- Tiene soporte para la Capa de Sockets Seguros (SSL, del inglés *Secure Sockets Layer*) y la Seguridad de la Capa de Transporte (TLS, del inglés *Transport Layer Security*), streaming de video, autenticación.
- Hosts virtuales: posee soporte para hosts virtuales basados en IP y/o hosts virtuales basados en nombre.
- Incluye servicios de correo electrónico con acceso al Protocolo de Acceso a Mensajes de Internet (IMAP, del inglés *Internet Message Access Protocol*) y al Protocolo de Oficina de Correo (POP, del inglés *Post Office Protocol*).
- Manejo de archivos estáticos, archivos de índices y autoindexado.
- Puede ser utilizado como proxy inverso con opciones de caché.
- Posee el módulo de reescritura de URL.
- Perl integrado.
- Logging personalizado.
- Control de acceso basado en la dirección IP del cliente y la autenticación básica de acceso HTTP.
- SSI (*Server Side Includes*), WebDAV limitado, transmisión FLV, FastCGI<sup>5</sup>.

---

<sup>5</sup> FastCGI: es un protocolo para interconectar programas interactivos con un servidor web. Es una variación de Interfaz Común de Entrada (CGI, del inglés *Common Gateway Interface*).

- Actualmente este servidor de páginas web es compatible con los principales Sistemas de Gestión de Contenidos (CMS, del inglés *Content Management System*) del mercado tales como Wordpress, Joomla, Drupal y phpBB, por lo que puede cumplir sin problemas con las necesidades de la mayor parte de los usuarios.

### 1.2.2 Configuración de Nginx

El servidor web Nginx se instala a través de la consola de Linux, en cualquier sistema operativo basado en GNU/Linux<sup>6</sup> en la ruta `/etc/nginx`. El archivo de configuración principal del mismo es `nginx.conf` y se puede encontrar en la siguiente ruta: `/etc/nginx/nginx.conf`. En el fichero `nginx.conf`, se encuentra la información referente a la configuración general del servidor, como son: la cantidad de procesos de trabajo que utiliza, las configuraciones del tráfico HTTP, las especificaciones de las directivas del servidor de correo y las directivas que afectan al procesamiento de la conexión. Los directorios principales de configuración son `sites-available`<sup>7</sup> y `sites-enabled`<sup>8</sup>, además de los ficheros `conf.d`, `koi-utf mime.types`, `naxsi.rules`, `scgi_params`, `win-utf`, `fastcgi_params`, `koi-win`, `naxsi_core.rules`, `naxsi-ui.conf.1.4.1`, `proxy_params`, `uwsgi_params`. En el directorio `sites-available` se crean los archivos de configuración específicos de cada host virtual disponible, y en el directorio `sites-enabled` se crean enlaces simbólicos a los archivos de configuración que se hayan creado en `sites-available`.

Una vez definidas las características de Nginx en cuanto a la administración, se hace necesaria la descripción de HMAST y los requerimientos que deben tener los módulos a integrar, pues constituye la base sobre la que se desarrolle el módulo propuesto.

### 1.3 Descripción de HMAST

HMAST es una aplicación web que permite la administración de máquinas servidoras. La administración se puede realizar tanto de forma local como remota, esta última a través de conexiones seguras mediante el uso del Protocolo Intérprete de Órdenes Seguro (SSH, del inglés *Secure Shell*). Tiene las funcionalidades necesarias para la gestión de usuarios y servicios telemáticos (Castillo, 2012). Las herramientas y tecnologías utilizadas en HMAST son: Spring como framework de desarrollo, Ajax para la

---

<sup>6</sup> GNU/Linux: sistema operativo, más conocido como Linux.

<sup>7</sup> `sites-available`: directorio que contiene los sitios disponibles en Nginx.

<sup>8</sup> `sites-enabled`: directorio que contiene los sitios habilitados en Nginx.

manipulación de contenidos en la web, IntelliJ IDEA como Entorno de Desarrollo Integrado (IDE, del inglés *Integrated Development Environment*), Visual Paradigm para Lenguaje Unificado de Modelado (UML, del inglés *Unified Modeling Language*) como Herramienta de Ingeniería de Software Asistida por Computadora (CASE, del inglés *Computer Aided Software Engineering*) y Git como sistema de control de versiones. Como metodología de desarrollo de software emplea Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) en su variación para la UCI.

### 1.3.1 Arquitectura de HMAST

La arquitectura de HMAST presenta un estilo N-Capas orientada al Dominio. Su estructura se basa en cinco capas (ver Figura 1) las cuales interactúan a través de interfaces y utilizando inyección de dependencias. Las características y principales responsabilidades de cada una de estas capas se describen a continuación.

- **Capa Presentación:** presenta al usuario los conceptos del negocio mediante una interfaz de usuario. Además, facilita la explotación de dichos procesos, informa sobre la situación de los procesos de negocio e implementación de las reglas de validación de dicha interfaz. En esta capa se realiza el tratamiento a las excepciones lanzadas desde capas inferiores.
- **Capa Aplicación:** realiza las llamadas a servicios de la capa inferior y tiene la responsabilidad de adaptar la información que recibe a los requisitos de los servicios de dominio. Está compuesta por los servicios de aplicación y los Objetos para la Transferencia de Datos (DTO, del inglés *Data Object Transfer*).
- **Capa Dominio:** se encarga de implementar la lógica de dominio, o sea, las reglas del negocio. Define las interfaces de persistencia a datos, conocidos como contratos a repositorios, pero no los implementa. Es responsable de realizar las validaciones y sus componentes solo dependen de la capa Infraestructura Transversal. Esta capa está compuesta por las entidades del dominio, los servicios de dominio y los contratos de repositorio. Las entidades representan objetos del dominio y están definidas fundamentalmente por su identidad y continuidad en el tiempo. Los servicios de dominio contienen la lógica que trata a las entidades como un todo y los contratos de repositorios son interfaces que especifican las operaciones que deben implementar los repositorios.

- **Capa Persistencia:** es responsable de contener el código necesario para persistir los datos. Se compone fundamentalmente por los repositorios, que son clases que implementan los contratos de repositorios definidos en la capa de Dominio.
- **Capa Infraestructura Transversal:** es responsable de promover la reutilización de código. Contiene las operaciones de seguridad, inicios de sesión, monitoreo del sistema, mecanismos de persistencia reutilizables, validadores genéricos y todas aquellas operaciones que se puedan utilizar desde otras capas (de la Torre, 2010).

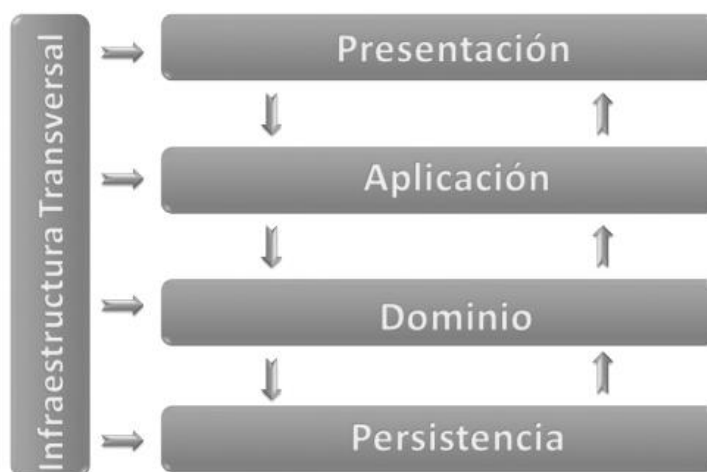


Figura 1: Arquitectura de HMAST (elaboración propia).

### 1.3.2 Funcionalidades que ofrece HMAST

La herramienta cuenta con funcionalidades tales como:

- Gestión de servidores lógicos: permite la adición, edición y eliminación de los datos de un servidor lógico, además permite la conexión remota y desconexión a un servidor seleccionado.
- Gestión de servicios telemáticos asociados a un servidor lógico: permite la adición, edición y eliminación de los datos de un módulo, así como activación y desactivación de los mismos.

- Gestión de las variables de configuración asociadas a un servidor lógico: permite cargar y salvar las variables de configuración de los servicios telemáticos encontrados en un servidor lógico (ficheros de configuración, nombre de módulos, demonios).

### **1.3.3 Consideraciones para implementar un módulo para HMAST**

Para la integración de un módulo a HMAST se deben tener en cuenta las siguientes consideraciones:

- La lógica de Aplicación no deberá incluir ninguna lógica del Dominio, solo tareas de coordinación relativas a requerimientos técnicos de la aplicación, como conversiones de formatos de datos de entrada a entidades del Dominio y llamadas a componentes de Infraestructura para que realicen tareas complementarias.
- La capa de Presentación no debe tener como entrada o salida, objetos de dominio, sino DTO.
- Las entidades solo pueden tener dependencias de componentes de la capa de Dominio.
- Las clases de servicios deben ser las únicas responsables de acceder a los repositorios, no se puede implementar código de persistencia a datos en la capa de Dominio.
- Solo se puede acceder a la información almacenada en los servidores haciendo uso de los repositorios.
- El código reutilizable por más de un repositorio debe estar disponible en la capa de Infraestructura Transversal (Palma, 2013).

Una vez definidas las características de HMAST, posteriormente se hace un estudio de las herramientas que administran Nginx, para determinar si cumplen con los requisitos para ser integradas.

### **1.4 Herramientas que administran Nginx**

Actualmente en la UCI, la administración de Nginx se realiza de forma manual, lo que se contrapone con las tendencias actuales de automatizar estos procesos. Aunque no son muchas, ya existen herramientas que automatizan la administración del servidor web Nginx, como es el caso de NginxTray y el módulo Nginx de Webmin, las cuales se describen a continuación.



### NginxTray

NginxTray es una herramienta privativa, desarrollada para ser una pequeña herramienta que permite manejar fácilmente el servidor web Nginx con un icono de la bandeja. Realiza funciones tales como: iniciar, detener y reiniciar el servidor y en cuanto a la configuración, iniciar y detener el Pre-procesador de Hipertexto (PHP, del inglés *Hypertext Preprocessor*) FastCGI, definir el directorio PHP FastCGI y establecer el nombre de archivo .exe y la dirección PHP.

### Módulo Nginx de Webmin

El módulo Nginx en la herramienta Webmin posee funcionalidades que se describen a continuación, ver Figura 2.

- Capacidad de deshabilitar y habilitar hosts virtuales: la columna *Status* muestra si el host virtual está habilitado o deshabilitado. Para deshabilitar o habilitar un host virtual, se puede marcar la casilla debajo de *Select Servers* y usar el menú desplegable *Select action*.
- Mostrar el nombre del fichero: la columna *Server Name* muestra el nombre del fichero que contiene el host virtual, al hacer clic se abrirá el host virtual para su edición.
- Mostrar la raíz del fichero: la columna *Document Root* muestra el directorio raíz del host virtual.
- Permitir una forma sencilla de acceder a la URL del host virtual: la columna *URL* proporciona un hipervínculo para abrir el sitio web en una nueva pestaña.
- Aplicar cambios: luego de actualizar los servidores se debe dar clic en la opción de aplicar cambios (*Apply Changes*).

Se puede utilizar la pestaña *Global Configuration* para modificar el archivo `/etc/nginx/nginx.conf`.

En la pestaña *Create Virtual Host* se pueden crear nuevos hosts virtuales.



Figura 2: Configuración del módulo Nginx en la herramienta Webmin (Mike, 2016).

## Resultado del estudio de homólogos

A pesar de las facilidades que brinda la herramienta NginxTray, presenta requisitos para ser utilizada que no son compatibles con Nova Servidores, la distribución cubana de GNU/Linux. El principal de ellos es que es una aplicación netamente privativa, a utilizar en sistemas operativos de la familia de Microsoft. Esta limitante de la herramienta trae consigo restringir su dominio de uso, dejando fuera de su alcance los servidores montados en entornos GNU/Linux. Webmin posee deficiencias, por lo que no se garantiza su correcto funcionamiento en las empresas cubanas y el módulo que posee para la administración del servidor web Nginx, no cumple con las características para ser integrado a HMAST. Ninguna de las dos herramientas analizadas anteriormente cumple con la arquitectura de HMAST ni con la forma de administración remota. Por otra parte, estas herramientas no permiten modificar en su totalidad todas las directivas del servidor, sino que lo hacen de forma limitada. Se decide entonces crear un módulo para HMAST que permita dar solución a estas limitaciones.

## 1.5 Herramientas y tecnologías

El módulo a desarrollar debe ser integrado a HMAST, por consecuente, las herramientas y tecnologías que se emplean, se deben corresponder con las mismas de la herramienta principal.

### ✓ Lenguaje de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y

semánticas que definen su estructura y el significado de sus elementos, respectivamente (Saavedra, 2007).

**Java:** es un lenguaje de programación orientado a objetos que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo. Los antecedentes de Java habría que buscarlos en los lenguajes de programación C y C++ (Rodríguez, 2006).

### ✓ **Framework**

Un framework (marco de trabajo) puede verse como una gran librería o conjunto de librerías donde además de facilitar funciones para su uso, se suele disponer de una sintaxis o metalenguaje específico del framework y una forma de organización del código específica (Krall, 2006).

**Spring:** permite el desarrollo de todo tipo de aplicaciones de manera rápida aplicando conceptos de modularización y facilita la realización de pruebas al código. Abstrae al desarrollador de los conceptos de “solicitudes” y “respuestas” y garantiza la reutilización de código dentro del software (González, 2016).

### ✓ **AJAX**

Para la manipulación de contenidos en la web se utilizará JavaScript Asíncrono y XML (AJAX, del inglés *Asynchronous JavaScript And XML*), que no es una tecnología por sí misma, sino un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Esto incluye: HTML o Lenguaje de Marcado para Hipertexto Extensible (XHTML, del inglés *eXtensible HyperText Markup Language*), Hojas de Estilo en Cascada (CSS, del inglés *Cascading Style Sheets*), JavaScript, Modelo de Objetos del Documento (DOM, del inglés *Document Object Model*), Lenguaje de Marcado Extensible (XML, del inglés *Xtensible Markup Language*), Lenguaje de Hojas de Estilo Extensible (XSLT, del inglés *EXTensible Stylesheet Language*), y el objeto XMLHttpRequest (Eguíluz, 2008). Cuando estas tecnologías se combinan en un modelo AJAX, es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. Esto crea aplicaciones más rápidas y con mejor respuesta a las acciones del usuario.

### ✓ **Augeas**

Augeas es una herramienta de edición de ficheros de configuración del sistema GNU/Linux. Analiza archivos de configuración en sus formatos nativos y los transforma en un árbol. Los cambios de configuración se realizan manipulando este árbol y guardándolo en archivos de configuración nativos (Elatov, 2014). Augeas es una herramienta modular, pero a diferencia de otras herramientas modulares esta depende totalmente de sus módulos, debido a que son los encargados de permitir la transformación de un fichero en árbol y viceversa. A estos módulos se le llaman Lenses y son los encargados de brindar soporte a cada fichero de configuración, pues cada uno describe un archivo (Tain, 2009). Esta herramienta se utiliza para la edición del fichero de configuración principal de Nginx *nginx.conf*.

### ✓ **Nginx-java-parser**

Es un analizador de la configuración del servidor web Nginx basado en la gramática Otra Herramienta para el Reconocimiento de Lenguajes (ANTLR, del inglés *ANother Tool for Language Recognition*) versión 4. Esta biblioteca ayuda a analizar los archivos de configuración, buscando parámetros especificados, bloques, expresiones regulares o comentarios (Plotnik, 2017). Se utiliza para la edición de los ficheros de los hosts virtuales.

### ✓ **Entorno de Desarrollo Integrado**

Un IDE es una aplicación que facilita la programación en un lenguaje, suele incluir un editor, herramientas de generación de código automática, de corrección sintáctica, de ejecución de código y depuración.

**IntelliJ IDEA:** es un IDE para el lenguaje de programación Java. Es ligero en el diseño y ofrece características útiles como pruebas de Junit, depuración, inspección de código y soporte para múltiple refactorización. IntelliJ IDEA se encuentra enfocado en elevar la productividad de los desarrolladores por lo que ofrece soporte avanzado para los frameworks y estándares más importantes en el desarrollo web como son Spring Framework, Vaadin, Play, Grails, Web Services y Struts. Además, incluye la asistencia de código para lenguajes como HTML, CSS, JavaScript, Node.js y TypeScript. Favorece la integración que tiene el IDE con herramientas para la construcción de proyectos como Maven, Ant y Gradle; y para el control de versiones como Git, SVN y Mercurial. Incorpora también un editor de base de datos SQL con soporte completo SQL para Oracle, PostgreSQL, MySQL y SQL Server (González, 2016).

### • **Lenguaje Unificado de Modelado**

UML es un lenguaje estándar utilizado para escribir planos de software. Se puede utilizar para visualizar, especificar, construir y documentar los artefactos y las relaciones entre ellos, que se crean durante el desarrollo del software. Se emplea para el diseño del módulo, específicamente para la construcción de los diagramas de clases y el diagrama de despliegue.

- **Herramienta de Ingeniería de Software Asistida por Computadora**

Una herramienta CASE es una herramienta individual para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento) (González, 2016).

**Visual Paradigm para UML:** es una herramienta CASE diseñada para una amplia gama de usuarios, incluyendo ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de forma fiable de los sistemas de software a gran escala con un enfoque orientado a objetos. Ofrece soporte para varias versiones de UML y para operaciones de ingeniería inversa de código a modelo o diagrama y de modelo o diagrama a código (González, 2016).

- **Sistema de control de versiones**

Un sistema de control de versiones es una herramienta que hará un seguimiento de los cambios en los archivos y permitirá la coordinación de diferentes desarrolladores que trabajan en partes de su sistema al mismo tiempo (Pilone, 2008). En el proyecto donde se realiza el módulo se utiliza Git por políticas de seguridad informática de la universidad.

**Git:** es un proyecto de código abierto, mantenido activamente, desarrollado en 2005 por Linus Torvalds, el famoso creador del núcleo del sistema operativo Linux. Un número asombroso de proyectos de software se basan en Git para el control de versiones, incluyendo proyectos comerciales, así como de código abierto. Los desarrolladores que han trabajado con Git están bien representados en el grupo de talento de desarrollo de software disponible y funciona bien en una amplia gama de sistemas operativos y IDEs (Atlassian, 2017).

- **ForeUI**

Es una herramienta de creación de prototipos de interfaz de usuario diseñada para crear mockup<sup>9</sup>, wireframe<sup>10</sup> o prototipos para cualquier aplicación o sitio web. Permite diseñar el comportamiento de prototipos definiendo diagramas de flujo intuitivos para manejar eventos específicos. ForeUI funciona en sistemas Windows, Mac OS X y Linux (EaSynth, 2015).

### **1.6 Metodología de desarrollo de software**

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Una determinada metodología no es necesariamente aplicable a todo tipo de proyectos, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor. Para el desarrollo del módulo para la administración del servidor web Nginx, con el fin de lograr la homogeneidad del sistema se emplea la metodología de desarrollo de software AUP-UCI, resultante de una variación de la metodología AUP que es la utilizada por la herramienta principal.

La UCI desarrolló una versión de la metodología de desarrollo de software AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma, y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, denominada Ejecución, y agregándose también una nueva fase denominada Cierre (Sánchez, 2015).

Los roles definidos por esta metodología son: jefe de proyecto, planificador, analista, arquitecto de información (opcional), desarrollador, administrador de la configuración, cliente/proveedor de requisitos, administrador de calidad, probador, arquitecto de sistema y administrador de base de datos. Además, AUP-UCI define cuatro escenarios para modelar el sistema. Se emplea el escenario número 4, el cual indica que proyectos que no modelen negocio solo pueden modelar el sistema con HU (Sánchez, 2015).

### **Conclusiones parciales**

La revisión de la bibliografía especializada en el servidor web Nginx, propició la comprensión de las características, funcionamiento, configuraciones y administración del mismo. La definición de los

---

<sup>9</sup> Mockup: fotomontajes que permiten a los diseñadores gráficos y web mostrar al cliente la propuesta de sus diseños.

<sup>10</sup> Wireframe: esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web.

requerimientos para integrar un módulo a HMAST, permitió identificar que las herramientas existentes para administrar Nginx, no solucionan el problema de la presente investigación. Además, el estudio de HMAST precisó que, para la integración del módulo, se debe emplear la metodología de desarrollo de software AUP-UCI y las herramientas: Spring, Ajax, Augeas, IntelliJ IDEA, Visual Paradigm, Git y ForeUI.

### Capítulo 2: Análisis y diseño del módulo Nginx

Una vez analizadas las características y funcionalidades del servidor web Nginx y mencionados los requerimientos que se necesitan en HMAST, en el presente capítulo se describe la propuesta del módulo a desarrollar. Se definirán las características y funcionalidades que tendrá el módulo mediante los requisitos funcionales y las Historias de Usuario. Además, se especifica la arquitectura y patrones de diseño a emplear.

#### 2.1 Propuesta de solución

La presente investigación propone el desarrollo de un módulo para la administración del servidor web Nginx desde HMAST. El módulo permitirá administrar el servicio de forma remota en PC servidoras mediante conexiones seguras utilizando el protocolo SSH, a través de una interfaz gráfica de usuario como se muestra en la Figura 3. El módulo contará con las funcionalidades necesarias para configurar las conexiones en el servidor, controlar el tráfico HTTP, gestionar los hosts virtuales, configurar los permisos de acceso y alias para cada host virtual.

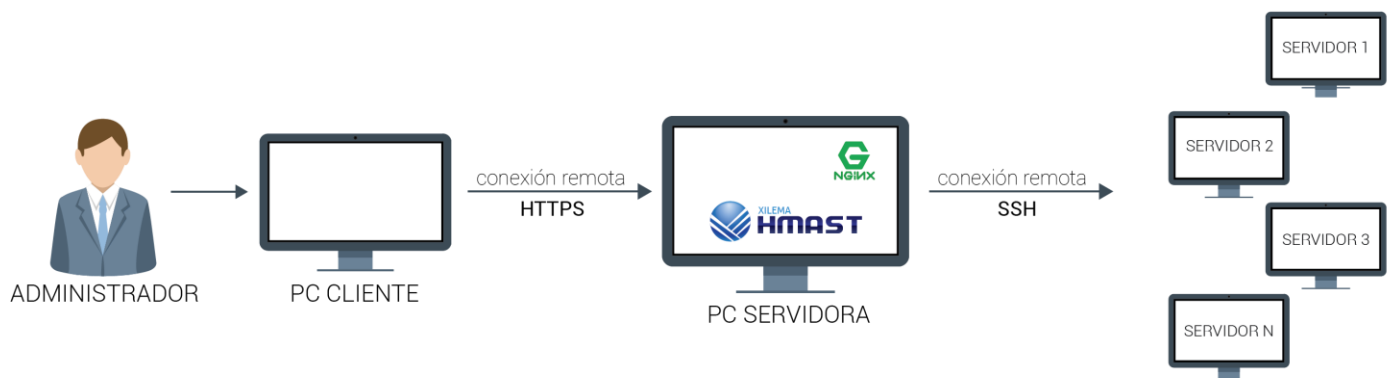


Figura 3: Propuesta de solución (elaboración propia).

#### 2.2 Artefactos generados

El proceso de desarrollo del módulo guiado por la metodología AUP-UCI genera como principal artefacto las Historias de Usuario. Teniendo en cuenta que no se modela el negocio, se ajustan las funcionalidades que se describen en un documento de Especificación de Requisitos de Software al escenario 4 que



## Capítulo 2: Análisis y diseño del módulo Nginx

establece esta metodología, el cual se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido, además se recomienda en proyectos no muy extensos, puesto que una HU no debe poseer demasiada información (Sánchez, 2015).

### 2.2.1 Especificación de requisitos

En ingeniería del software y el desarrollo de sistemas, un requisito es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requisitos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario. Muestran qué elementos y funciones son necesarias para un proyecto (Alegsa, 2016).

Los requisitos funcionales describen las funciones que el software debe ejecutar, con el objetivo de establecer un entendimiento común entre el usuario y el proyecto de software. La siguiente tabla muestra el listado de los requisitos funcionales del módulo, clasificados según su prioridad en alta, media y baja.

Tabla 1. Listado de requisitos funcionales.

No.	Nombre del requisito funcional	Descripción
<b>Prioridad</b>		<b>Alta</b>
RF1	Instalar el servidor web Nginx	Permite instalar el servidor web Nginx.
RF2	Desinstalar el servidor web Nginx.	Permite desinstalar el servidor web Nginx.
RF3	Iniciar el servidor web Nginx	Permite iniciar el servidor web Nginx.
RF4	Detener el servidor web Nginx	Permite detener el servidor web Nginx.
RF5	Reiniciar el servidor web Nginx	Permite reiniciar el servidor web Nginx.
RF6	Recargar el servidor web Nginx	Permite recargar el servidor web Nginx.
RF7	Adicionar host virtual	Permite adicionar un host virtual a Nginx.
<b>Prioridad</b>		<b>Media</b>

## Capítulo 2: Análisis y diseño del módulo Nginx

RF8	Mostrar conexiones	Permite mostrar las conexiones del servidor web Nginx.
RF9	Editar conexiones	Permite editar las conexiones del servidor web Nginx.
RF10	Listar hosts virtuales	Permite listar todos los hosts virtuales asignados a Nginx.
RF11	Editar host virtual	Permite editar un host virtual.
RF12	Eliminar host virtual	Permite eliminar un host virtual.
RF13	Mostrar host virtual	Permite mostrar los parámetros de un host virtual.
RF14	Habilitar host virtual	Permite habilitar un host virtual.
RF15	Deshabilitar host virtual	Permite deshabilitar un host virtual.
<b>Prioridad</b>		<b>Baja</b>
RF16	Listar directorio	Permite listar los directorios.
RF17	Adicionar directorio	Permite adicionar un directorio.
RF18	Editar directorio	Permite editar un directorio.
RF19	Mostrar directorio	Permite mostrar un directorio.
RF20	Eliminar directorio	Permite eliminar un directorio.
RF21	Mostrar las configuraciones del tráfico HTTP	Permite mostrar las configuraciones del tráfico HTTP.
RF22	Editar las configuraciones del tráfico HTTP	Permite editar las configuraciones del tráfico HTTP.
RF23	Aplicar cambios al servidor	Permite aplicar los cambios realizados localmente en la PC servidora.

## *Capítulo 2: Análisis y diseño del módulo Nginx*

RF24	Descartar cambios locales	Permite descartar los cambios locales realizados en la PC donde está instalada HMAST.
------	---------------------------	---

Los requisitos no funcionales describen atributos solo del sistema o del ambiente del sistema que no están relacionados directamente con los requisitos funcionales. Incluyen restricciones cuantitativas, como el tiempo de respuesta o precisión, tipo de plataforma (lenguajes de programación y/o sistemas operativos).

*Tabla 2: Listado de requisitos no funcionales (elaboración propia).*

No	Nombre del requisito no funcional	Atributo de calidad	Descripción
1	El módulo se ejecutará sobre el sistema operativo GNU/Linux Nova Servidores.	Funcionalidad	Estos requisitos son heredados de HMAST, con los que debe cumplir el módulo para ser integrado a dicha herramienta.
2	Emplear como lenguaje de programación Java.		
3	Disponer en el sistema operativo GNU/Linux de los siguientes paquetes: augeas-tools, libjna-java, openjdk-7-jdk.		
4	Proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o modificar los mismos, y las personas o sistemas autorizados tengan el acceso a ellos.	Seguridad (Acceso restringido)	Todos los datos manejados por el módulo estarán encriptados; tanto los que se transfieren entre la estación cliente y el servidor HMAST como los que se almacenan temporalmente en el servidor.
5	Permitir al usuario aprender su aplicación.	Usabilidad	Internacionalizar la información que se muestra, en los idiomas español e inglés

## Capítulo 2: Análisis y diseño del módulo Nginx

6	El módulo debe mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interfaz especificada.	Confiabledad (Tolerancia a fallos)	Ante el fallo de una funcionalidad del sistema el resto de las funcionalidades que no dependen de esta, deberán seguir funcionando.
---	--	------------------------------------	---

### 2.2.2 Historias de Usuario

Las Historias de Usuario especifican las tareas que debe realizar el sistema, lo que equivale a los casos de uso en el Proceso Unificado. Son escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto. Guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo (González, 2016).

A continuación, se muestran las Historias de Usuario: Instalar el servidor web Nginx, Adicionar host virtual, Listar hosts virtuales, Editar conexiones y Eliminar host virtual. Las restantes se encuentran en los Anexos.

Tabla 3: HU Instalar el servidor web Nginx (elaboración propia).

<b>Número:</b> 1	<b>Nombre del requisito:</b> Instalar el servidor web Nginx.		
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1		
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 20 horas		
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 10 horas		
<b>Descripción:</b> el sistema debe permitir al usuario instalar el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema, si el servidor no está instalado, puede instalarlo.			
<b>Observaciones:</b> en la PC servidora se ejecuta el comando <code>apt-get install nginx</code> .			
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>			

Módulo nginx

### Módulo nginx



### Descripción del servicio

Servidor web HTTP de código abierto, es un servidor proxy de correo y genérico TCP/UDP. Se utiliza para publicar contenido estático. Es conocido por su alto rendimiento, estabilidad, conjunto de características ricas, configuración sencilla y bajo consumo de recursos. Es ligero, multiplataforma y fácil de instalar. Además posee soporte para host virtuales basados en ip y/o basados en nombre.

### Datos del servicio

Nombre del servicio	nginx
Paquetes a instalar	nginx
Demonio	nginx
Directorio principal	/etc/nginx
Fichero principal	/nginx.conf
Directorio de los hosts virtuales disponibles	/sites-available
Directorio de los hosts virtuales habilitados	/sites-enabled

[⚙️ Instalar](#)

Tabla 4: HU Adicionar host virtual.

<b>Número:</b> 7	<b>Nombre del requisito:</b> Adicionar host virtual.		
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1		
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 20 horas		
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 20 horas		
<p><b>Descripción:</b> el sistema debe permitir al usuario adicionar un host virtual a Nginx. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente presiona el botón Adicionar para que muestre la interfaz de adición, la cual tiene los parámetros agrupados en tres pestañas: Generales, Directorios y Otros parámetros. Los campos a especificar se encuentran en la primera y última pestaña, estos son:</p> <ul style="list-style-type: none"> <li>Nombre del fichero: se refiere al nombre del fichero que contiene la configuración del host virtual. Es un campo obligatorio. Debe estar conformado por no más de 255 caracteres.</li> <li>Dirección IP: se refiere a la dirección IP del host virtual en caso de existir más de una interfaz de red. Responde a la primera parte del argumento de la directiva <i>listen</i>. Es un campo obligatorio. La dirección IP puede ser un asterisco (*) en caso de que se defina acceder a un host virtual desde cualquier dirección IP con un puerto determinado. Cuando la directiva <i>listen</i> aparece solo con el</li> </ul>			

valor del puerto, por defecto asume la dirección IP de la PC servidora.

- Puerto: se refiere al puerto de escucha asignado al host virtual, que debe ser un número entero positivo entre 1 y 65535. Responde a la segunda parte del argumento de la directiva `listen`. Es un campo obligatorio. El puerto puede ser un asterisco (\*) en caso de que se defina acceder a un host virtual mediante una dirección especificada por cualquier puerto.
- Directorio raíz: se refiere al directorio que almacena el contenido a publicar por el host virtual. Debe ser la ruta de un directorio existente en la PC servidora. Es el argumento de la directiva `root`. Es un campo obligatorio.
- Nombres del servidor: se refiere a la lista de nombres de dominio asignados al host virtual. Cada elemento debe ser un nombre de dominio válido. Es el argumento de la directiva `server_name`. No es un campo obligatorio.
- Archivos índices: se refiere a la lista de nombres de los archivos índices del directorio raíz del host virtual. Es el argumento de la directiva `index`. No es un campo obligatorio.

**Observaciones:** una vez introducidos dichos parámetros se crea en la PC local el fichero de configuración del host.

### Prototipo elemental de la interfaz gráfica de usuario:

Adicionar host virtual

Generales Directorios Otros parámetros

Nombre del fichero: \*  ?

Dirección IP: \*  ?

Puerto: \*  ?

Directorio raíz: \*  ?

Cancelar Aceptar

Detailed description: The image shows a graphical user interface window titled 'Adicionar host virtual'. It has a close button (X) in the top right corner. Below the title bar, there are three tabs: 'Generales' (selected), 'Directorios', and 'Otros parámetros'. The 'Generales' tab contains four input fields, each with a red asterisk indicating it is required. The fields are: 'Nombre del fichero: \*', 'Dirección IP: \*', 'Puerto: \*', and 'Directorio raíz: \*'. Each field has a small question mark icon to its right. The 'Directorio raíz' field has a folder icon to its right. At the bottom right of the window, there are two buttons: 'Cancelar' and 'Aceptar'.

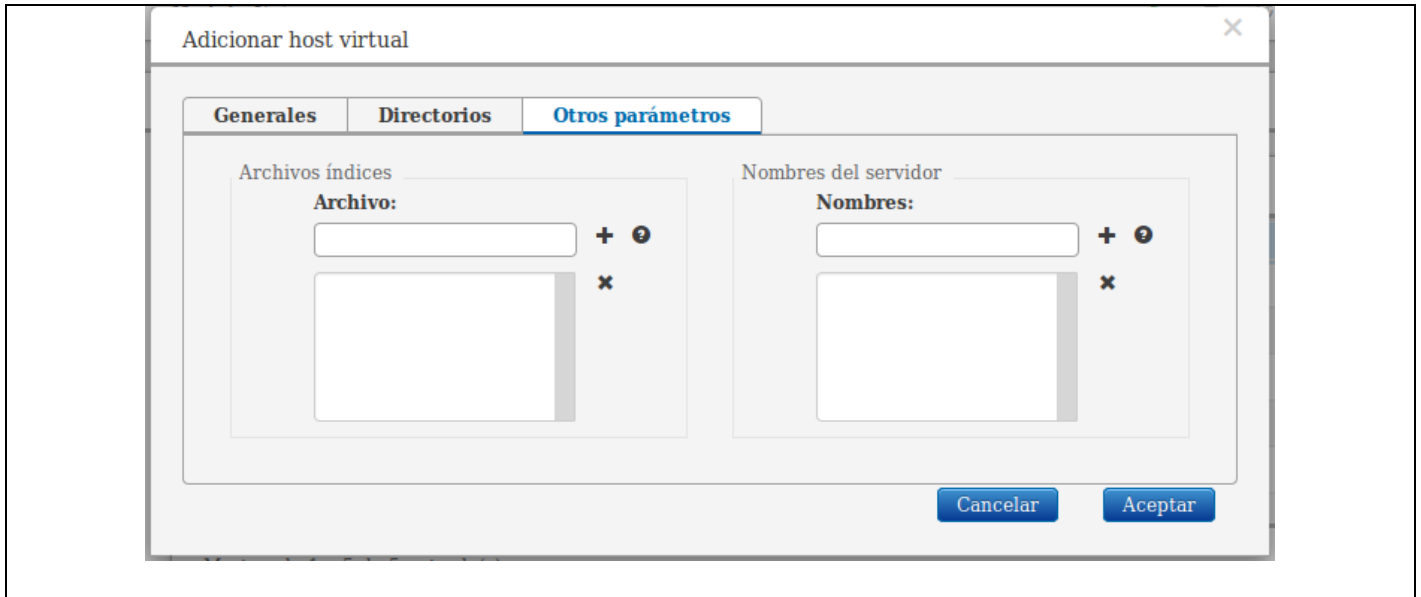


Tabla 5: HU Listar host virtual.

<b>Número:</b> 10	<b>Nombre del requisito:</b> Listar host virtual.	
<b>Programador:</b> Rachel Molina Rodríguez		<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media		<b>Tiempo Estimado:</b> 20 horas
<b>Riesgo en desarrollo:</b> Medio		<b>Tiempo Real:</b> 20 horas
<b>Descripción:</b> el sistema debe permitir al usuario mostrar el listado de todos los hosts virtuales existentes en el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. En el caso que no hayan hosts virtuales asignados a Nginx, el sistema muestra un mensaje notificándolo.		
<b>Observaciones:</b>		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		

## Capítulo 2: Análisis y diseño del módulo Nginx

Hosts Virtuales					
Buscar <input type="text"/>					Mostrar 5 entradas
	Ficheros	Dirección IP	Puerto	Directorio raíz	Habilitado
<input type="checkbox"/>	host2.com.conf	10.53.3.231	9093	/var/www/host2.com/html	Si
<input type="checkbox"/>	default	10.53.3.231	85	/var/www/html	No
<input type="checkbox"/>	host1.com.conf	10.53.3.231	9092	/var/www/host1.com/html	No
<input type="checkbox"/>	ejemplo.com.conf	10.53.3.231	80	/var/www/ejemplo.com/html	Si
<input type="checkbox"/>	ejemplo.com.conf	10.53.3.231	9090	/var/www/ejemplo2.com/html	No

Mostrando 1 a 5 de 5 entrada(s)

Tabla 6: HU Editar conexiones.

<b>Número:</b> 9	<b>Nombre del requisito:</b> Editar conexiones.	
<b>Programador:</b> Rachel Molina Rodríguez		<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media		<b>Tiempo Estimado:</b> 20 horas
<b>Riesgo en desarrollo:</b> Medio		<b>Tiempo Real:</b> 10 horas
<p><b>Descripción:</b> el sistema debe permitir al usuario especificar los parámetros referentes a las conexiones en Nginx. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Conexiones. Se pueden editar los parámetros siguientes:</p> <ul style="list-style-type: none"> <li>• <b>Tiempo entre peticiones subsiguientes:</b> se refiere a la cantidad de segundos que esperará el servidor en atender una nueva petición antes de cerrar la conexión. Es obligatorio y debe ser un número entero entre 0 y 2147483647. El valor por defecto es 75 segundos. Responde a la directiva <i>keepAlive_TimeOut</i>.</li> <li>• <b>Cantidad de procesos de trabajo:</b> se refiere a la cantidad máxima de procesos de trabajo en el servidor. Es obligatorio y debe ser un número entero entre 0 y 2147483647. La cantidad de procesos de trabajo debe ser menor o igual que la cantidad de procesos disponibles en el CPU de</li> </ul>		



## Capítulo 2: Análisis y diseño del módulo Nginx

la PC. El valor por defecto es 1 proceso. Responde a la directiva *worker\_processes*. Si el valor de dicha directiva es *auto*, significa que el servidor intenta detectar de forma automática el número de núcleos del CPU disponibles en la PC.

- Cantidad de conexiones simultáneas por procesos de trabajo: se refiere a la cantidad máxima de conexiones simultáneas por procesos aceptadas por el servidor. Es obligatorio y debe ser un número entero entre 0 y 2147483647. El valor por defecto es 512. Responde a la directiva *worker\_connections*.
- Un proceso de trabajo acepta conexiones persistentes: se refiere a si el servidor acepta o no varias peticiones por conexión, con argumento on/off que significa que el servidor acepta conexiones persistentes (on) o no acepta conexiones persistentes (off). El valor por defecto es off. Responde a la directiva *multi\_accept*. Si *multi\_accept* está deshabilitado, un proceso de trabajo aceptará una conexión a la vez, de lo contrario aceptará todas las conexiones nuevas a la vez.

**Observaciones:** el argumento *auto* es compatible a partir de las versiones 1.3.8 y 1.2.5. Esos parámetros se encuentran en el fichero *nginx.conf* de la PC servidora, que está almacenado localmente.

### Prototipo elemental de la interfaz gráfica de usuario:

Conexiones			
Directiva	Valor	Editar	Detalles
Tiempo entre peticiones subsiguientes	65		
Cantidad de procesos de trabajo	1		
Cantidad de conexiones simultáneas por procesos de trabajo	768		
Un proceso de trabajo acepta conexiones persistentes	No		

Tabla 7: HU Eliminar host virtual.

Número: 12	Nombre del requisito: Eliminar host virtual.
------------	--

## Capítulo 2: Análisis y diseño del módulo Nginx

<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 15 horas
<b>Riesgo en desarrollo:</b> Medio	<b>Tiempo Real:</b> 15 horas
<b>Descripción:</b> el sistema debe permitir al usuario eliminar el/los hosts virtuales deseados. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. En el caso que existan hosts virtuales asignados a Nginx, el usuario selecciona el/los que desea eliminar y presiona el botón Eliminar. Posteriormente se muestra un mensaje de confirmación. En caso positivo, se eliminan dichos hosts virtuales y se muestra un mensaje notificando si la eliminación fue satisfactoria o si hubo errores.	
<b>Observaciones:</b> El fichero se elimina del directorio que se encuentra de forma local.	
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>	
 <p>The screenshot shows a web application window titled "Hosts Virtuales". At the top right, there are window control buttons: a plus sign, a refresh icon, a close button (circled in green), and two other icons. Below the title bar, there is a search bar labeled "Buscar" and a "Mostrar" dropdown menu set to "5" with the text "entradas" to its right. The main content is a table with the following columns: "Ficheros", "Dirección IP", "Puerto", "Directorio raíz", and "Habilitado". Each column has a small diamond icon to its right. The table contains five rows of data. The first row has a checkbox, "host2.com.conf", "10.53.3.231", "9093", "/var/www/host2.com/html", and "Si". The second row has a checkbox, "default", "10.53.3.231", "85", "/var/www/html", and "No". The third row has a checkbox, "host1.com.conf", "10.53.3.231", "9092", "/var/www/host1.com/html", and "No". The fourth row has a checked checkbox, "ejemplo.com.conf", "10.53.3.231", "80", "/var/www/ejemplo.com/html", and "Si". The fifth row has a checkbox, "ejemplo.com.conf", "10.53.3.231", "9090", "/var/www/ejemplo2.com/html", and "No". At the bottom of the table, there is a status bar that says "Mostrando 1 a 5 de 5 entrada(s)" and two circular navigation arrows.</p>	

### 2.3 Arquitectura del módulo

El diseño de la arquitectura de un sistema es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define los componentes que forman el sistema, su relación entre ellos y la funcionalidad especificada que llevan a cabo mediante su interacción,

cumpliendo los criterios de calidad indicados como seguridad, disponibilidad, eficiencia y usabilidad (de la Torre, 2010). El proceso de desarrollo del módulo utiliza la arquitectura establecida para HMAST, para lograr la estabilidad del sistema, ver Figura 4.



Figura 4: Arquitectura del módulo (elaboración propia).

Se emplea una arquitectura N-Capas orientada al dominio, que tiene como objetivo estructurar la complejidad de una aplicación empresarial basada en las diferentes capas de la arquitectura siguiendo el patrón N-Capas y las tendencias de arquitectura orientadas al dominio (González, 2016). El paquete del módulo propuesto se inserta en las capas donde se realizará su implementación; en la capa de Infraestructura Transversal es donde estarán todas las clases reutilizables.

### 2.4 Diagrama de paquetes

Para el diseño del diagrama de paquetes se toma como referencia la arquitectura propuesta, donde existe un paquete correspondiente a cada una de las capas de la arquitectura, y dentro de cada uno de ellos, un paquete llamado *Nginx* que contiene las clases pertenecientes al módulo.

La Figura 5 representa el diagrama de paquetes con las diferentes capas de la arquitectura del módulo. En la capa de Presentación está el paquete *nginx* que contiene la clase controladora *Nginx* que se encarga de implementar los métodos para el envío y recepción de datos para cada una de las vistas del módulo. En la capa de Aplicación, se encuentra el paquete *nginx* que contiene la clase *INginxAppService* donde se definen los métodos que se llamarán desde la capa de Presentación y la clase *NginxAppService*

## Capítulo 2: Análisis y diseño del módulo *Nginx*

que es la encargada de la implementación de los métodos de la interfaz mencionada anteriormente. En *nginx* se encuentra también el paquete *dto*, que contiene los DTO, los cuales son enviados desde y hacia la capa de Presentación, que depende de la interfaz *INginxAppService*. La clase encargada de convertir los DTO en entidades y viceversa es *NginxAppService* en la capa de Aplicación.

La capa de Dominio se relaciona con la capa de Aplicación a través de una relación de agregación con la clase *INginxAppService* mediante la inyección de dependencias proporcionada por el framework Spring. Dentro de esta capa se encuentra el paquete *nginx* que contiene los paquetes *service*, *repositorycontrats* y *entities*. En *service* se define la interfaz *INginxService* y la clase *NginxService* que es donde se implementan los métodos definidos en dicha interfaz, además, tiene una relación de agregación con la clase *INginxRepository* mediante la inyección de dependencias. En *repositorycontrats* se definen los contratos de repositorio en la interfaz *INginxRepository*. En *entities* se definen las entidades del módulo, que representan los objetos del dominio.

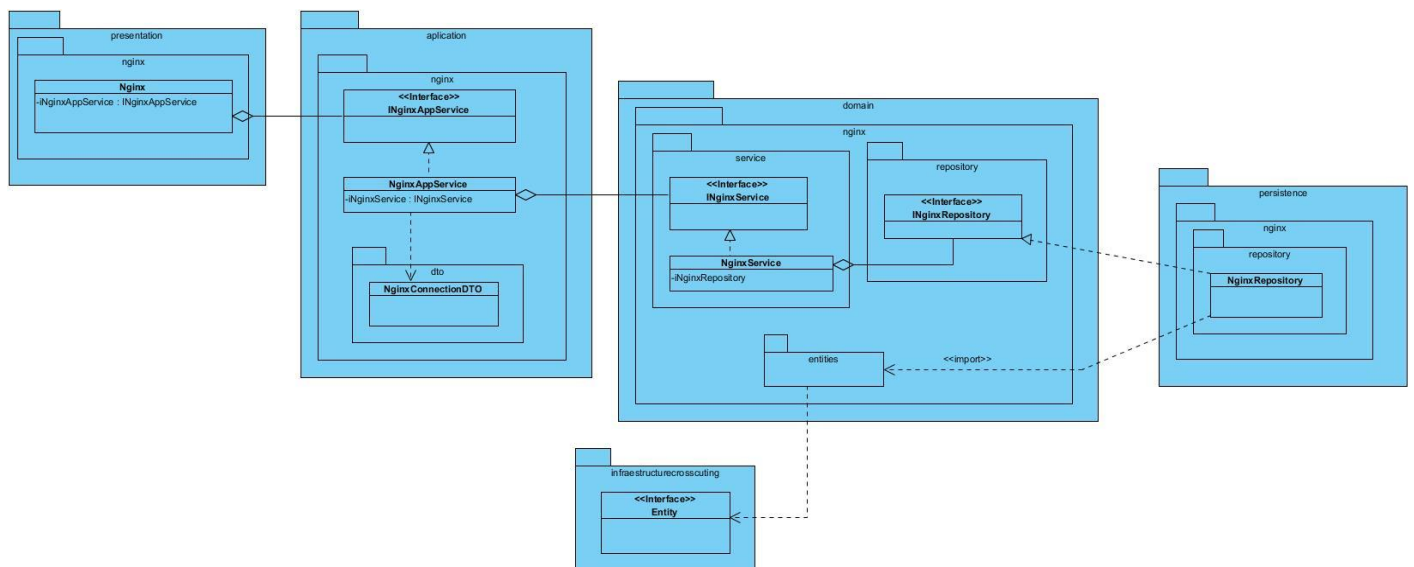


Figura 5: Diagrama de paquetes.

En la capa de Persistencia se representa el paquete *nginx* que contiene un paquete llamado *repository*, en el cual se encuentra la clase *NginxRepository* que se encarga de implementar los métodos de la interfaz *INginxRepository*, con el objetivo de lograr la persistencia de los datos que se manejan en el módulo.

### 2.4.1 Diagrama de clases de las entidades

La Figura 6 representa el diagrama de clases de las entidades pertenecientes al paquete *entities* de la capa de Dominio, donde se encuentran las entidades siguientes: VirtualHost, Connection, HostName, DirectoryIndex, Location, DirectoryAlias, Access, HttpTraffic.

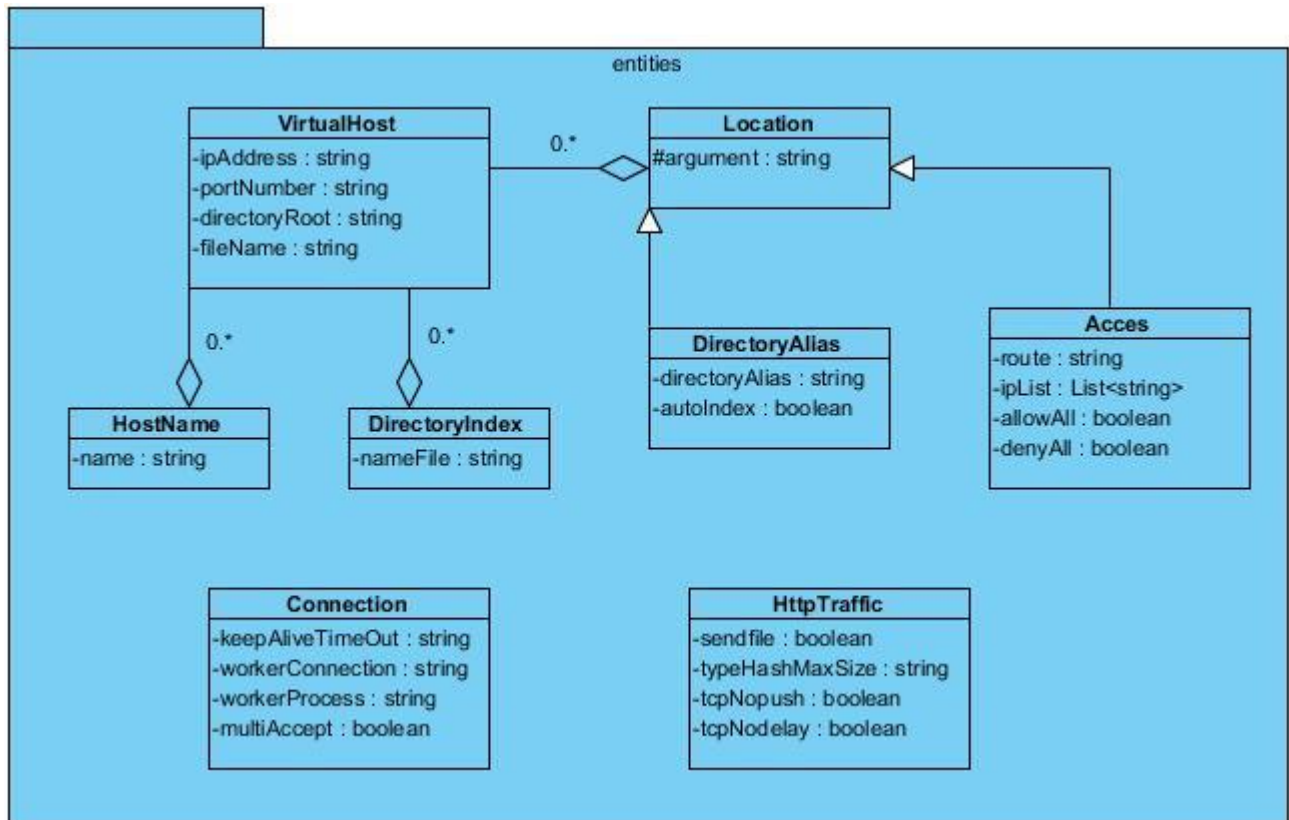


Figura 6: Diagrama de clases de las entidades.

### 2.5 Patrones de diseño

Los patrones de diseño son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (González, 2016). En el modelo del diseño del módulo se aplican los Patrones Generales de Software para Asignar Responsabilidades (GRASP, del inglés *General Responsibility Assignment Software Patterns*) y GoF (del inglés *Gang of Four*).

### 2.5.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. A continuación, se describen los patrones utilizados.

**Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos. Se utiliza en la capa de Aplicación en la clase *NginxAppService* al realizarse la conversión de clases de tipo DTO a entidades, pues se crean instancias de las entidades debido a que las clases DTO contienen los datos de inicialización de estas. Además, se utiliza en la conversión de entidades a DTO.

**Experto:** se utiliza para la asignación de responsabilidades relacionadas con la obtención de información. Este patrón se manifiesta en todas las capas, pues cada una tiene un objetivo y una responsabilidad asignada, siendo cada capa la encargada de implementarla.

**Bajo Acoplamiento:** consiste en asignar responsabilidades de modo que el acoplamiento permanezca bajo. El mecanismo de inyección de dependencias proporcionado por el framework Spring, permite el uso de este patrón. Además, permite la reutilización de las clases y que no sean afectadas por cambios que se realicen en otros componentes. Este patrón se emplea en las distintas capas del módulo mediante el uso de interfaces que relacionan una capa con otra de forma que dichas relaciones no se establezcan directamente hacia las clases. Las conexiones se realizan a través del mecanismo de inyección de dependencias. Esto se evidencia en el paquete *repositorycontrats*, donde se define la interfaz *INginxRepository*, y la implementación de sus métodos es realizada por la clase *NginxRepository* en la capa de Persistencia.

**Alta Cohesión:** la cohesión es una medida del grado de focalización de las responsabilidades de una clase. Permite que las clases sean fáciles de entender, mantener y reutilizar. Se manifiesta en las validaciones en cada una de las entidades específicamente en el método *Validate*.

### 2.5.2 Patrones GoF

Los patrones GoF son patrones de asignación de responsabilidades y se pueden categorizar en tres grupos teniendo en cuenta su propósito: creacionales, estructurales y de comportamiento.

**Patrón Solitario (Singleton):** es un patrón de tipo creacional con el objetivo de garantizar la existencia de una única instancia para una clase y posibilitar el acceso global a dicha instancia. Su empleo se evidencia cuando se realiza una conexión SSH a un servidor, pues con una única instancia del objeto conexión se realizan las operaciones sobre este. Tal es el caso de la clase Connections, donde con una única instancia se realizan todas las operaciones en el servidor.

### **Conclusiones parciales**

El estudio realizado en el Capítulo 1 permitió definir las características y funcionalidades del módulo propuesto, obteniéndose 24 requisitos funcionales y 6 requisitos no funcionales. La arquitectura propuesta, se tomó como base para el diseño del diagrama de paquetes, permitiendo describir los patrones de diseño utilizados.

### Capítulo 3: Implementación y pruebas al módulo Nginx

En el presente capítulo se exponen los estándares de codificación empleados para el desarrollo del módulo, con el objetivo de mantener la uniformidad con la codificación de los restantes módulos integrados a la herramienta. Se establece la estrategia de pruebas para verificar la calidad del módulo implementado y se documenta la realización de las pruebas seleccionadas.

#### 3.1 Estándar de codificación

Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto permite que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea entendible (Arias, 2009). Los estándares de codificación que se emplean en el módulo se ajustan a las pautas definidas en el expediente de proyecto de HMAST.

- **Asignación de nombres:** emplear descriptores en inglés. 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 *CamelCase*<sup>11</sup>.
- **Ficheros de código fuente:** cada fichero contiene una única clase o interfaz. Si hay una clase privada o una interfaz asociada a una clase pública se puede poner en el mismo fichero. La clase pública debe ser la primera.
- **Indentación:** la unidad de indentación de bloques de sentencias son 4 espacios.
- **Comentarios:** los comentarios deben añadir claridad al código. Deben contar el por qué y no el cómo. Deben ser concisos.
- **Declaraciones:** se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

---

<sup>11</sup> *CamelCase* es la práctica de escribir frases o palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra.



- **Continuidad de las líneas largas:** cuando una sentencia no quepa en una única línea se debe fraccionar después de una coma, después de un operador y alinear la nueva línea con el comienzo de la expresión al mismo nivel de la anterior.
- **Longitud de la línea:** limitar todas las líneas a un máximo de 120 caracteres.
- **Nombres de componentes:** todos los paquetes comienzan con `cu.uci.hmast.xx.yy.zz.kk`

xx → presentation, application, domain, persistence.

yy → nombre del módulo (*nginx*).

zz → elementos que pueden contener los componentes verticales (*entities, repositorys*).

kk → clases o subpaquetes.

Posteriormente a la implementación del módulo, se procede a realizarle pruebas para comprobar su correcto funcionamiento y el cumplimiento de la propuesta de solución descrita en el capítulo anterior.

### 3.2 Estrategia de pruebas

Según Pressman, una estrategia de prueba se puede ver como una espiral, comenzando con la prueba de unidad, la cual se centra en el código fuente de cada unidad del software, luego la prueba de integración donde el objeto de atención es el diseño y la construcción de la arquitectura del software, posteriormente se encuentra la prueba de validación donde se validan los requisitos establecidos como parte del análisis de requisitos del software, y por último la prueba del sistema en el contexto de ingeniería del sistema (Pressman, 2009).

Esta estrategia tiene un enfoque incremental y se representa mediante una espiral que analiza el código, el diseño, los requisitos y la ingeniería del sistema para lo cual se emplean pruebas de unidad, de integración, de validación y de sistema, respectivamente. En la Figura 7 se representa la estrategia de pruebas descrita anteriormente.

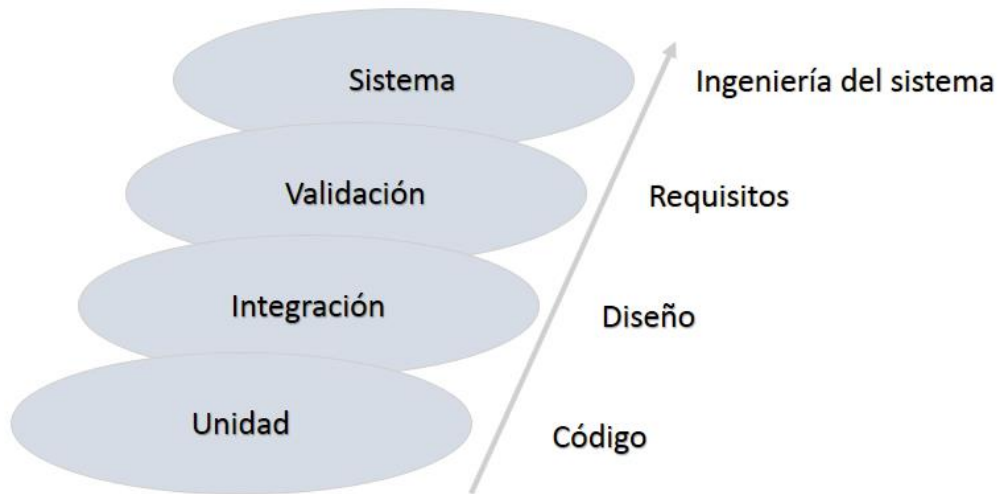


Figura 7: Estrategia de pruebas de software (Pressman, 2009).

La prueba de unidad analiza el módulo asegurando que funciona como una unidad, haciendo un uso extensivo de las técnicas de prueba de caja blanca, ejercitando caminos específicos de la estructura de control del módulo para asegurar un alcance completo y una detección máxima de errores. Durante la prueba de integración, las técnicas que más prevalecen son las de diseños de casos de prueba de caja negra y se pueden llevar a cabo algunas pruebas de caja blanca. La prueba de validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento, donde se usan exclusivamente técnicas de prueba de caja negra. La prueba del sistema verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total (Pressman, 2009).

La metodología AUP-UCI divide las pruebas en tres disciplinas: internas, liberación y aceptación. Las pruebas de liberación son diseñadas y ejecutadas por una entidad certificadora externa, por lo que en el presente trabajo no son analizadas.

### 3.2.1 Pruebas internas

Para la aplicación de las pruebas internas la autora se basa en las pruebas de unidad e integración.

### Prueba de unidad

La prueba de unidad que se realiza hace uso de la técnica de prueba de caja blanca del camino básico. El método del camino básico permite al diseñador de casos de pruebas obtener una medida de la complejidad lógica de un diseño procedural y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (Pressman, 2009).

Las unidades de prueba más pequeñas son las operaciones dentro de la clase. A continuación, se realiza la técnica del camino básico, abarcando el código en todas las capas a la operación *updateConnections*, que permite editar las conexiones del servidor web Nginx.

Tabla 8: Código de la funcionalidad *updateConnections* .

(1)	<code>public void updateConnections(UUID idLogicalServer, ConnectionsDTO connectionsDTO) throws EEntityNotExist, EInvalidEntity, InterruptedException, ENotFoundUrlRemoto, NoSuchAlgorithmException, IOException, JSchException, SftpException, EFileWithIncorrectSettingsForAugeas, EInvalidProcess {</code>
(2)	<code>Connections connections = new Connections(connectionsDTO.getId(), connectionsDTO.getKeepAliveTimeOut(), connection sDTO.getWorkerProcesses(), connectionsDTO.getWorkerConnections(), connectionsDTO.isM ultiAccept());</code>
(3)	<code>LogicalServer server=serversAdministrationService.getLogica lServerById(idLogicalServer);</code>
(4)	<code>ValidationErrors validationErrors = new ValidationErrors();</code>
(5)	<code>connections.validate(validationErrors);</code>
(6)	<code>if (validationErrors.hasErrors()) {</code>
(7)	<code>throw new EInvalidEntity(validationErrors);</code>
(8)	<code>}</code>
(9)	<code>if(NginxValidator.workerProcessesInteger(connections.getWorkerProcesses()) &gt; NginxValidator.workerProcessesInteger(iNginxRepository.cpuProcesses(server))) {</code>
(10)	<code>throw new</code>

	<code>EInvalidProcess("nginx.connections.workerprocesses.invalidprocess");</code>
(11)	<code>}</code>
(12)	<code>else{</code>
(13)	<code>  this.connectionsHashMap.remove(idServer);</code>
(14)	<code>  this.connectionsHashMap.put(idServer, connection);</code>
(15)	<code>  saveNginxConnectionConfiguration(server, connections);</code>
(16)	<code>}</code>
(17)	<code>}</code>

Luego de numerar las líneas de código, se diseña la gráfica del programa que describe el flujo de control lógico empleando nodos y aristas, como se aprecia en la Figura 8.

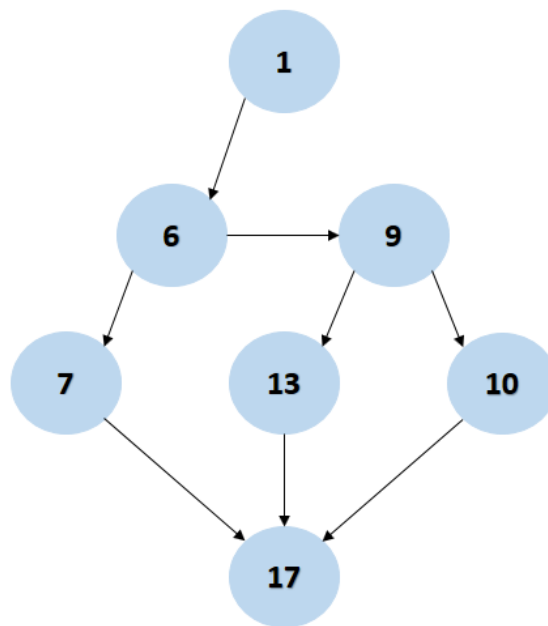


Figura 8: Grafo de flujo (elaboración propia).

A partir del grafo obtenido con 7 nodos y 8 aristas, se calcula la complejidad ciclomática  $V(G)$  la cual constituye una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa (Pressman, 2009).

$$V(G) = \text{cantidad\_aristas} - \text{cantidad\_nodos} + 2$$

$$V(G) = 8 - 7 + 2 = 3$$

Un camino independiente es cualquier camino del programa que introduce al menos un nuevo conjunto de sentencias de proceso o una nueva condición (Pressman, 2009). La cantidad de caminos independientes se establece por la complejidad ciclomática, por tanto, se identifican 3 caminos como se muestra en la siguiente tabla.

*Tabla 9: Listado de caminos independientes.*

No.	Camino
1	1-6-7-17
2	1-6-9-10-17
3	1-6-9-13-17

El valor de la complejidad ciclomática ofrece además un límite superior para la cantidad de pruebas que se deben diseñar y ejecutar para garantizar que se cumplen todas las sentencias del programa (Pressman, 2009). Por tanto, se diseñan casos de prueba para cada camino independiente.

*Tabla 10: Caso de prueba de unidad para el camino 1.*

<b>Caso de prueba de unidad</b>	
<b>No. Camino:</b> 1	<b>Camino:</b> 1-6-7-17
<b>Nombre de la persona que realiza la prueba:</b> Rachel Molina Rodríguez	
<b>Descripción de la prueba:</b> Verificar la edición de las conexiones del servidor web Nginx.	
<b>Entrada:</b> Se envía como parámetro el valor de cada una de las conexiones siguientes: Tiempo entre peticiones subsiguientes: 738989899999999 Cantidad de procesos de trabajo: 10 Cantidad de conexiones simultáneas por procesos de trabajo: 778 Un proceso de trabajo acepta conexiones persistentes: Sí	
<b>Resultado esperado:</b> El sistema debe mostrar un mensaje de error si el valor de alguno de estos	

## Capítulo 3: Implementación y pruebas al módulo Nginx

parámetros no está entre 0 y 2147483647, el sistema marca en rojo el campo incorrecto y la función de aceptar no se activa hasta que no estén correctos los valores.
<b>Evaluación de la prueba:</b> Satisfactoria. Se obtuvo el mensaje “La cantidad de conexiones subsiguientes no es válida”.

Tabla 11: Caso de prueba de unidad para el camino 2.

Caso de prueba de unidad	
<b>No. Camino:</b> 2	<b>Camino:</b> 1-6-9-10-17
<b>Nombre de la persona que realiza la prueba:</b> Rachel Molina Rodríguez	
<b>Descripción de la prueba:</b> Verificar la edición de las conexiones del servidor web Nginx.	
<b>Entrada:</b> Se obtuvo la cantidad de procesos disponibles en el CPU y es 1 proceso. Se envía como parámetro el valor de cada una de las conexiones siguientes: Tiempo entre peticiones subsiguientes: 73 Cantidad de procesos de trabajo: 10 Un proceso de trabajo acepta conexiones persistentes: 778 Un proceso de trabajo acepta conexiones persistentes: Sí	
<b>Resultado esperado:</b> El sistema debe mostrar un mensaje de error ya que la cantidad de procesos de trabajo especificada es mayor que la cantidad de procesos disponibles en el CPU, en este caso mayor que 1.	
<b>Evaluación de la prueba:</b> Satisfactoria. Se obtuvo el mensaje “La cantidad de procesos de trabajo es mayor que los procesos disponibles en el CPU”.	

Tabla 12: Caso de prueba de unidad para el camino 3.

Caso de prueba de unidad	
<b>No. Camino:</b> 3	<b>Camino:</b> 1-6-9-13-17
<b>Nombre de la persona que realiza la prueba:</b> Rachel Molina Rodríguez	
<b>Descripción de la prueba:</b> Verificar la edición de las conexiones del servidor web Nginx.	
<b>Entrada:</b> Se envía como parámetro el valor de cada una de las conexiones siguientes: Tiempo entre peticiones subsiguientes: 78	

Cantidad de procesos de trabajo: 1
Cantidad de conexiones simultáneas por procesos de trabajo: 625
Un proceso de trabajo acepta conexiones persistentes: No
<b>Resultado esperado:</b> El sistema debe mostrar un mensaje satisfactorio al modificar estas conexiones.
<b>Evaluación de la prueba:</b> Satisfactoria. Se obtuvo el mensaje “Las conexiones han sido editadas correctamente”.

En todos los casos de prueba aplicados se obtuvieron resultados satisfactorios, demostrándose, por tanto:

- ✓ La ejecución de cada sentencia del código al menos una vez.
- ✓ La ejecución de cada condición en sus vertientes verdadera y falsa.
- ✓ La validez de las estructuras internas de datos.

### Prueba de integración

La prueba de integración es una técnica sistemática para construir la estructura del software. El objetivo es tomar componentes probados mediante la prueba de unidad y construir una estructura de programa de acuerdo con el diseño (Pressman, 2009). Asumiendo la arquitectura por capas empleada y que el módulo debe integrarse a una herramienta base, se emplea una estrategia de integración ascendente, donde empieza la construcción y las pruebas a los componentes de los niveles más bajos de la estructura del programa hacia arriba.

En el contexto de una estrategia de prueba de integración ascendente, se realiza la prueba de regresión, que permite volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos colaterales no deseados (Pressman, 2009).

#### 3.2.2 Pruebas de aceptación

En el caso de las pruebas de aceptación se aplican las de validación según la estrategia planteada por Pressman, con el objetivo de validar que el sistema cumple con el funcionamiento esperado y permitir que el cliente determine su aceptación desde el punto de vista de su funcionalidad y rendimiento. Tienen el

objetivo de verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales fue construido (Sánchez, 2015).

Para el cumplimiento de las pruebas de aceptación, el cliente realizó una revisión al sistema con el objetivo de verificar la implementación y el funcionamiento de cada uno de los requisitos funcionales definidos, emitiendo un acta de aceptación en total conformidad con la aplicación, ver anexo 3.

### **Conclusiones parciales**

El uso de los estándares de codificación definidos para HMAST permitió desarrollar un código reutilizable y fácil de entender. La elección de una estrategia de pruebas con un enfoque incremental propició comprobar el módulo en cada una de sus partes y las relaciones entre ellas. Con la realización de las pruebas se verificó que todas las instrucciones del módulo se ejecutan al menos una vez, que los componentes se integran correctamente, se validó que el módulo se ajusta al sistema y cumple con las exigencias del cliente.



## **Conclusiones generales**

- El desarrollo de un módulo para administrar el servidor web Nginx en HMAST, puede garantizar la adaptabilidad de la herramienta al entorno empresarial cubano.
- El módulo permite el despliegue de la herramienta HMAST en ecosistemas en los cuales se emplee el servidor web Nginx.
- La automatización de la administración de Nginx, apoya el proceso de migración a software libre y plataformas de código abierto.
- Las diferentes configuraciones personalizables del módulo permiten su adaptación a las disímiles necesidades de la empresa cubana.

## **Recomendaciones**

Se recomienda para próximas versiones del producto que Nginx pueda ser configurado como proxy inverso desde el módulo.

## Referencias Bibliográficas

**Alegsa, Leandro.** *Alegsa.com.ar.* **2016.** [En línea]. Disponible en: <http://www.alegsa.com.ar/Dic/requerimientos.php>. [Accedido: 29-jun-2016].

**Arias Calleja, Manuel.** CISIAD. *CISIAD.* **2009.** [En línea]. Disponible: <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>. [Accedido: 27-mar-2017.]

Atlassian. Atlassian. *Atlassian.* **2017.** [En línea]. Disponible: <https://www.atlassian.com/git/tutorials/what-is-git>. [Accedido: 22-mar-2017].

**Bautista, Jose Gregorio.** slideshare. **2011.** [En línea] Disponible: <http://es.slideshare.net/josegregoriob/servidor-web-8451426>. [Accedido: 3-abr-2011].

**Camilo, Sol Angelita Mercedes.** blogs EOI. *blogs EOI.* **2012.** [En línea] Disponible: <http://www.eoi.es/blogs/solangelitacamilo/2012/01/30/el-internet-y-el-mundo/> [Accedido: 30-jun-2012].

**Castillo, Reidiel y P. Soria.** Herramienta para la Migración y Administración de Servidores (HMAS). Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, junio **2012.**

**de la Torre, César, y otros.** *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0.* s.l. : Krasis PRESS, **2010.**

**DragonE.** Codeplex. *Codeplex.* **2014.** [En línea] Disponible: <https://nginxtray.codeplex.com/> [Accedido: 25-nov-2016].

**EaSynth Solution Inc. Ltd.** ForeUI. *ForeUI.* **2015.** [En línea] Disponible: <http://www.foreui.com/> [Accedido: 10-dic-2016].

**Eguíluz Pérez, Javier.** Introducción a AJAX. **2008.** [En línea]. Disponible en: [http://dis.um.es/~lopezquesada/documentos/IES\\_1314/IAW/curso/UT7/libroswebajax/www.librosweb.es/ajax/pdf/introduccion\\_ajax.pdf](http://dis.um.es/~lopezquesada/documentos/IES_1314/IAW/curso/UT7/libroswebajax/www.librosweb.es/ajax/pdf/introduccion_ajax.pdf). [Accedido: 30-abr-2017].

**Elatov, Karim.** [En línea] **2014.** Disponible: <http://elatov.github.io/2014/09/using-augeas-to-modify-configuration-files/> [Accedido: 15-02-2016].

**González, Felipe y Vera, Yosel.** *MÓDULO DE HMAST PARA LA ADMINISTRACIÓN Y MIGRACIÓN HACIA SAMBA4 DEL SERVICIO DIRECTORIO ACTIVO.* Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, junio, 2016.

**IEEE-STD-830-1998: PRÁCTICA RECOMENDADA PARA LAS ESPECIFICACIONES DE REQUISITOS DEL SOFTWARE.** 2008.

- Krall, César.** Aprender a programar. *Aprender a programar.* 2006. [En línea] Disponible: [http://www.aprenderaprogramar.com/index.php?option=com\\_attachments&task=download&id=803](http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=803). [Accedido: 25-mar-2016].
- Mike.** Guides. *Guides.* 2016. [En línea] Disponible: <https://guides.wp-bullet.com/using-webmin-module-to-manage-nginx-sites-on-easyengine/> [Accedido: 28-mayo-2017].
- Netcraft.** NETCRAFT. *NETCRAFT.* 2017. [En línea] Disponible: <https://news.netcraft.com/archives/2017/03/24/march-2017-web-server-survey.html>. [Accedido: 3-abr-2017.]
- Palma, Nurisel.** *Módulo para la administración de los servidores web en HMAST.* Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, junio, 2013.
- Parada, Miguel.** Ubuntizando. *Ubuntizando.* 2012. [En línea] Disponible: <http://www.ubuntizando.com/pencil-aplicacion-libre-y-muy-interesante-para-la-creacion-de-interfaces-graficos/>. [Accedido: 7-may-2017].
- Pilone, Dan y Miles, Russ.** Head First Software Development. [En línea] 26 de marzo de 2008.
- Plotnik, Alexey.** Github. *Github.* 2017. [En línea] Disponible: <https://github.com/odiszapc>. [Accedido: 30-mar-2017].
- Pressman, Roger.** 2009. Estrategias de prueba del software. *Ingeniería del software. Un enfoque práctico.* 5ta. 2009, págs. 307,308.
- Rodríguez, Alex.** Aprender a programar. 2006. [En línea] Disponible: <http://www.aprenderaprogramar.com>. [Accedido: 8-mar-2017].
- Romero, Hermenegildo.** 2012. slideshare. [En línea] 7 de febrero de 2012.
- Saavedra, Jorge.** Lenguajes de Programación. *Lenguajes de Programación.* 2007. [En línea] Disponible: <https://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>. [Accedido: 5-may-2017].
- Sánchez, Tamara.** 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* Universidad de las Ciencias Informaticas. La Habana : s.n., 2015.
- Tain Domínguez, Yuliette y Valdivia Genó, Román Miguel.** 2009. *Augeas, propuesta tecnológica para la gestión de archivos de configuración de sistemas GNU/Linux.* [En línea] mayo de 2009.

**Anexo 1: Entrevista al administrador del Nodo Central en la UCI.**

**Nombre:** Yadier Perdomo Cuevas

**Preguntas:**

1. ¿Qué sitios publican su contenido utilizando el servidor web Nginx?
2. ¿Existe alguna herramienta en la UCI para administrar el servidor web Nginx?
3. ¿Cómo funciona el servidor web Nginx?
4. Si el servidor se administra de forma manual. ¿Se han planteado crear una herramienta para facilitar dicha administración?
5. ¿Cuáles son las configuraciones principales que se deben tener en cuenta en la administración del servidor web Nginx?

**Anexo 2: Historias de Usuario.**

*Tabla 13: HU Desinstalar el servidor web Nginx.*

<b>Número:</b> 2	<b>Nombre del requisito:</b> Desinstalar el servidor web Nginx.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> el sistema debe permitir al usuario desinstalar el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y en el listado de módulos, presiona la opción Desinstalar en el módulo Nginx.		
<b>Observaciones:</b> se ejecuta en la PC servidora el comando <i>apt-get remove nginx</i> y el comando <i>apt-get remove nginx-common</i> .		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		

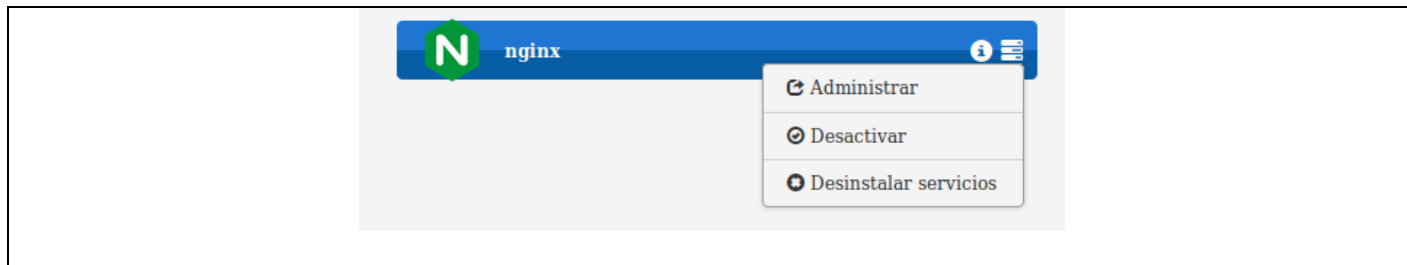


Tabla 14: HU Mostrar host virtual.

<b>Número:</b> 8	<b>Nombre del requisito:</b> Mostrar host virtual.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 20 horas	
<b>Riesgo en desarrollo:</b> Medio	<b>Tiempo Real:</b> 15 horas	
<p><b>Descripción:</b> el sistema debe permitir al usuario mostrar los parámetros de un host virtual. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente presiona el Editar para que muestre la interfaz con los parámetros (nombre del fichero, dirección IP, puerto, directorio raíz, archivos índices, nombres del servidor) de un host virtual en el servidor web Nginx.</p>		
<b>Observaciones:</b>		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		

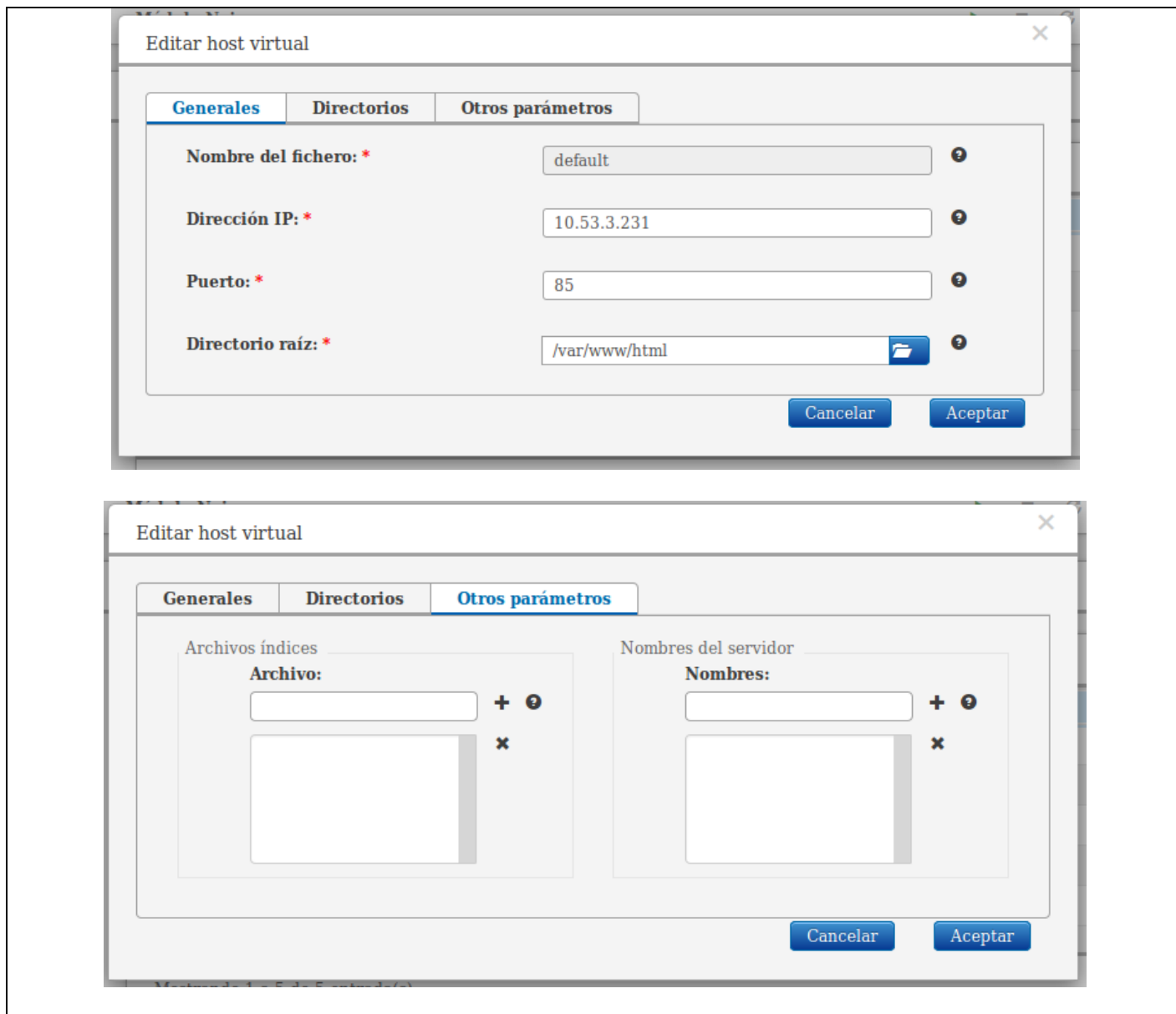


Tabla 15: HU Editar host virtual.

<b>Número:</b> 9	<b>Nombre del requisito:</b> Editar host virtual.
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1

<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 20 horas
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 20 horas
<p><b>Descripción:</b> el sistema debe permitir al usuario editar un host virtual en el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Debe existir al menos un host virtual en el listado, posteriormente selecciona uno y presiona el botón Editar para que muestre la interfaz de edición. Se muestra al usuario todas las opciones que tiene para modificar:</p> <ul style="list-style-type: none"> <li>• Nombre del fichero: este parámetro no se puede editar, se mantiene el nombre del fichero que posee el host virtual que va a ser editado.</li> <li>• Dirección IP: se refiere a la dirección IP del host virtual en caso de existir más de una interfaz de red. Responde a la primera parte del argumento de la directiva <i>listen</i>. Es un campo obligatorio. La dirección IP puede ser un asterisco (*) en caso de que se defina acceder a un host virtual desde cualquier dirección IP con un puerto determinado. Cuando la directiva <i>listen</i> aparece solo con el valor del puerto, por defecto asume la dirección IP de la PC servidora.</li> <li>• Puerto: se refiere al puerto de escucha asignado al host virtual, que debe ser un número entero positivo entre 1 y 65535. Responde a la segunda parte del argumento de la directiva <i>listen</i>. Es un campo obligatorio. El puerto puede ser un asterisco (*) en caso de que se defina acceder a un host virtual mediante una dirección especificada por cualquier puerto.</li> <li>• Directorio raíz: se refiere al directorio que almacena el contenido a publicar por el host virtual. Debe ser la ruta de un directorio existente en la PC servidora. Es el argumento de la directiva <i>root</i>. Es un campo obligatorio.</li> <li>• Nombres del servidor: se refiere a la lista de nombres de dominio asignados al host virtual. Cada elemento debe ser un nombre de dominio válido. Es el argumento de la directiva <i>server_name</i>. No es un campo obligatorio.</li> <li>• Archivos índices: se refiere a la lista de nombres de los archivos índices del directorio raíz del host virtual. Es el argumento de la directiva <i>index</i>. No es un campo obligatorio.</li> </ul>	



**Observaciones:** una vez introducidos dichos parámetros se edita en la PC local el fichero de configuración del host.

**Prototipo elemental de la interfaz gráfica de usuario:**

The screenshot shows a dialog box titled "Editar host virtual" with a close button (X) in the top right corner. It features three tabs: "Generales" (selected), "Directorios", and "Otros parámetros". The "Generales" tab contains four labeled input fields, each with a help icon (question mark) to its right:

- Nombre del fichero: \*** with the value "default".
- Dirección IP: \*** with the value "10.53.3.231".
- Puerto: \*** with the value "85".
- Directorio raíz: \*** with the value "/var/www/html" and a folder selection icon.

At the bottom right of the dialog are two buttons: "Cancelar" and "Aceptar".

The screenshot shows the same dialog box "Editar host virtual" but with the "Otros parámetros" tab selected. It is divided into two main sections:

- Archivos índices:** Contains a sub-section labeled "Archivo:" with a text input field, a "+" icon, and a help icon. Below it is a list box with an "x" icon for removal.
- Nombres del servidor:** Contains a sub-section labeled "Nombres:" with a text input field, a "+" icon, and a help icon. Below it is a list box with an "x" icon for removal.

At the bottom right of the dialog are two buttons: "Cancelar" and "Aceptar".

Tabla 16: HU Iniciar el servidor web Nginx.


<b>Número:</b> 3	<b>Nombre del requisito:</b> Iniciar el servidor web Nginx.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 5 horas	
<b>Descripción:</b> el sistema debe permitir al usuario iniciar el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y presiona el botón Iniciar.		
<b>Observaciones:</b> se ejecuta en la PC servidora el comando <i>service nginx start</i> . El servicio debe estar detenido.		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		
		

Tabla 17: HU Detener el servidor web Nginx.

<b>Número:</b> 4	<b>Nombre del requisito:</b> Detener el servidor web Nginx.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 5 horas	
<b>Descripción:</b> el sistema debe permitir al usuario detener el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y presiona el botón Detener.		
<b>Observaciones:</b> se ejecuta en la PC servidora el comando <i>service nginx stop</i> . El servicio debe estar corriendo.		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		



Tabla 18: HU Reiniciar el servidor web Nginx.

<b>Número:</b> 5	<b>Nombre del requisito:</b> Reiniciar el servidor web Nginx.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 5 horas	
<b>Descripción:</b> el sistema debe permitir al usuario reiniciar el servidor web Nginx. La funcionalidad comienza cuando el usuario accede al sistema y presiona el botón Reiniciar.		
<b>Observaciones:</b> se ejecuta en la PC servidora el comando <code>service nginx restart</code> . El servicio debe estar corriendo.		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		

Tabla 19: HU Recargar el servidor web Nginx.

<b>Número:</b> 6	<b>Nombre del requisito:</b> Recargar el servidor web Nginx.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Medio	<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> el sistema debe permitir al usuario recargar el servidor web Nginx. Esta acción se ejecuta		

automáticamente después de ejecutarse la HU 23, Aplicar cambios en la PC servidora.
<b>Observaciones:</b> se ejecuta el comando <i>service nginx reload</i> en la PC servidora.
<b>Prototipo elemental de la interfaz gráfica de usuario:</b> no tiene prototipo de interfaz

Tabla 20: HU Habilitar host virtual.

<b>Número:</b> 14	<b>Nombre del requisito:</b> Habilitar host virtual.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 15 horas	
<b>Riesgo en desarrollo:</b> Medio	<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> el sistema debe permitir al usuario habilitar un host virtual. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente el usuario debe marcar el/los que desea que estén habilitados y presionar el botón Habilitar. Se muestra un mensaje de confirmación. En caso positivo se habilitan los hosts virtuales seleccionados y se muestra un mensaje de notificación, ya sea satisfactoria o de error.		
<b>Observaciones:</b>		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		

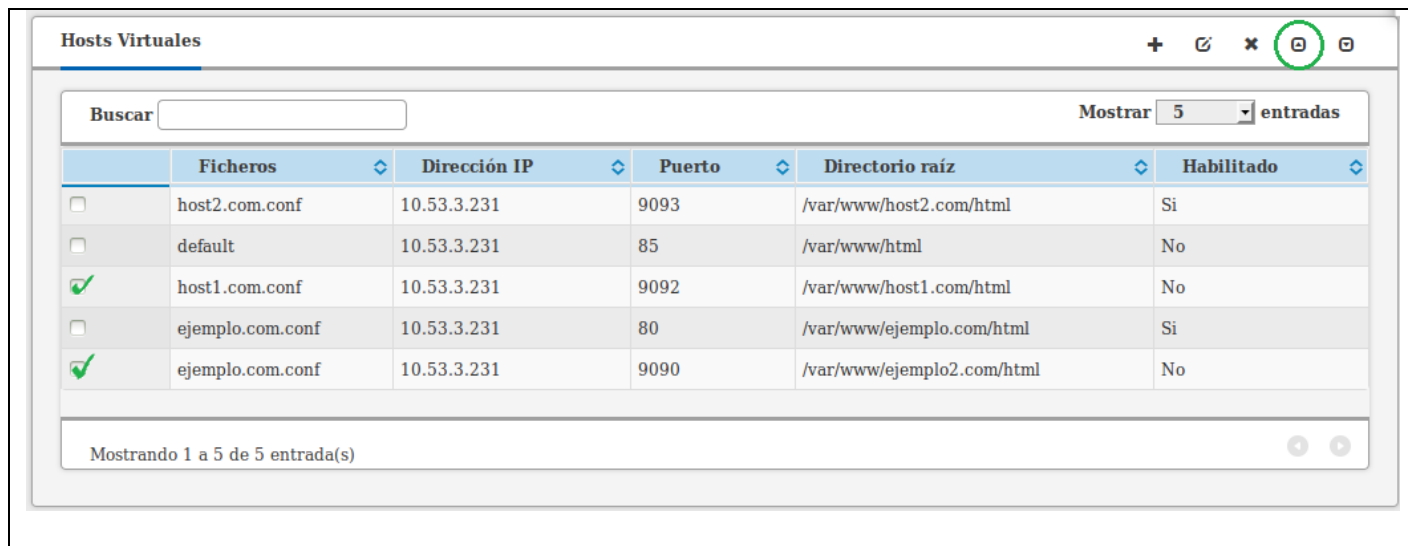


Tabla 21: HU Deshabilitar host virtual.

<b>Número:</b> 15	<b>Nombre del requisito:</b> Deshabilitar host virtual.		
<b>Programador:</b> Rachel Molina Rodríguez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Media		<b>Tiempo Estimado:</b> 15 horas	
<b>Riesgo en desarrollo:</b> Medio		<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> el sistema debe permitir al usuario deshabilitar un host virtual. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente el usuario debe marcar el/los que desea que estén deshabilitados y presionar el botón Deshabilitar. Se muestra un mensaje de confirmación. En caso positivo se deshabilitan los hosts virtuales seleccionados y se muestra un mensaje de notificación, ya sea satisfactoria o de error.			
<b>Observaciones:</b>			
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>			

	Ficheros	Dirección IP	Puerto	Directorio raíz	Habilitado
<input checked="" type="checkbox"/>	host2.com.conf	10.53.3.231	9093	/var/www/host2.com/html	Si
<input type="checkbox"/>	default	10.53.3.231	85	/var/www/html	No
<input type="checkbox"/>	host1.com.conf	10.53.3.231	9092	/var/www/host1.com/html	No
<input checked="" type="checkbox"/>	ejemplo.com.conf	10.53.3.231	80	/var/www/ejemplo.com/html	Si
<input type="checkbox"/>	ejemplo.com.conf	10.53.3.231	9090	/var/www/ejemplo2.com/html	No

Tabla 22: HU Mostrar Conexiones.

<b>Número:</b> 8	<b>Nombre del requisito:</b> Mostrar conexiones.		
<b>Programador:</b> Rachel Molina Rodríguez		<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Media		<b>Tiempo Estimado:</b> 15 horas	
<b>Riesgo en desarrollo:</b> Alto		<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> el sistema debe permitir mostrar los parámetros referentes a las conexiones. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Conexiones.			
<b>Observaciones:</b>			
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>			

Conexiones			
Directiva	Valor	Editar	Detalles
Tiempo entre peticiones subsiguientes	65		
Cantidad de procesos de trabajo	1		
Cantidad de conexiones simultáneas por procesos de trabajo	768		
Un proceso de trabajo acepta conexiones persistentes	No		

Tabla 23: HU Mostrar las configuraciones del tráfico HTTP.

<b>Número:</b> 21	<b>Nombre del requisito:</b> Mostrar las configuraciones del tráfico HTTP.																				
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1																				
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 15 horas																				
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 10 horas																				
<b>Descripción:</b> el sistema debe permitir mostrar las configuraciones del tráfico HTTP. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Configuración HTTP.																					
<b>Observaciones:</b>																					
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>																					
<div style="border: 1px solid #ccc; padding: 10px;"> <p><b>Configuración HTTP</b></p> <table border="1"> <thead> <tr> <th>Directiva</th> <th>Valor</th> <th>Editar</th> <th>Detalles</th> </tr> </thead> <tbody> <tr> <td>Tamaño máximo de los tipos de tablas hash</td> <td>1024</td> <td></td> <td></td> </tr> <tr> <td>Enviar fichero</td> <td>Si</td> <td></td> <td></td> </tr> <tr> <td>Opción TCP_NOPUSH</td> <td>No</td> <td></td> <td></td> </tr> <tr> <td>Opción TCP_NODELAY</td> <td>No</td> <td></td> <td></td> </tr> </tbody> </table> <p style="text-align: right;"> <input type="button" value="✕ Cancelar"/> <input type="button" value="✓ Aceptar"/> </p> </div>		Directiva	Valor	Editar	Detalles	Tamaño máximo de los tipos de tablas hash	1024			Enviar fichero	Si			Opción TCP_NOPUSH	No			Opción TCP_NODELAY	No		
Directiva	Valor	Editar	Detalles																		
Tamaño máximo de los tipos de tablas hash	1024																				
Enviar fichero	Si																				
Opción TCP_NOPUSH	No																				
Opción TCP_NODELAY	No																				

Tabla 24: HU Editar las configuraciones del tráfico HTTP.

<b>Número:</b> 22	<b>Nombre del requisito:</b> Editar las configuraciones del tráfico HTTP.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 15 horas	
<b>Riesgo en desarrollo:</b> Alto	<b>Tiempo Real:</b> 10 horas	
<p><b>Descripción:</b> el sistema debe permitir editar las configuraciones del tráfico HTTP. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Configuración HTTP. Se podrán editar los parámetros:</p> <ul style="list-style-type: none"> <li>• Tamaño máximo de los tipos de tablas hash: establece el tamaño máximo de los tipos de tablas hash. El valor por defecto es 1024. Responde a la directiva <i>type_hash_max_size</i>.</li> <li>• Enviar ficheros: activa o desactiva el envío de ficheros. El valor por defecto es <i>off</i>. Responde a la directiva <i>sendfile</i>.</li> <li>• Opción TCP-NOPUSH: activa o desactiva la opción de socket TCP-NOPUSH en FreeBSD o la opción de socket TCP-CORK en Linux. Las operaciones solo se activan cuando se utiliza el envío de ficheros. El valor por defecto es <i>off</i>. Responde a la directiva <i>tcp_nopush</i>.</li> <li>• Opción TCP-NODELAY: habilita o deshabilita la opción TCP-NODELAY. Solo se habilita cuando una conexión pasa a estado <i>keep_alive</i>. El valor por defecto es <i>off</i>. Responde a la directiva <i>tcp_nodelay</i>.</li> </ul>		
<b>Observaciones:</b>		
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>		



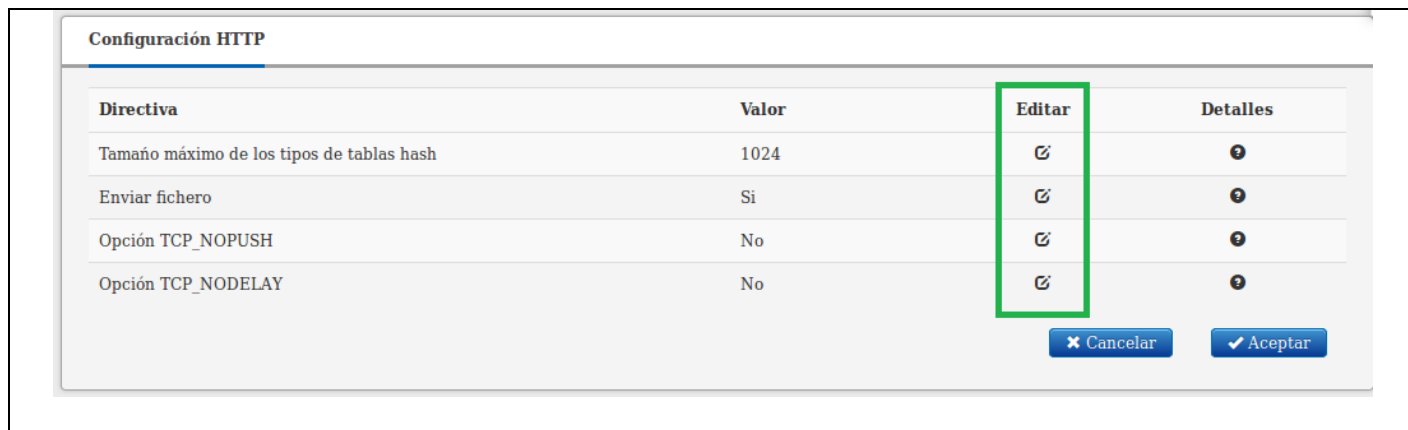


Tabla 25: HU Listar directorio.

<b>Número:</b> 16	<b>Nombre del requisito:</b> Listar directorio.		
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1		
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 20 horas		
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 20 horas		
<b>Descripción:</b> el sistema debe permitir al usuario listar los parámetros de alias y permisos de acceso pertenecientes a la directiva <i>location</i> . La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente puede presionar el botón Adicionar o el Editar, en ambas interfaces existe una pestaña destinada a la gestión de los directorios, donde se muestra la lista de los mismos o un mensaje notificando si no existe ninguno.			
<b>Observaciones:</b>			
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>			

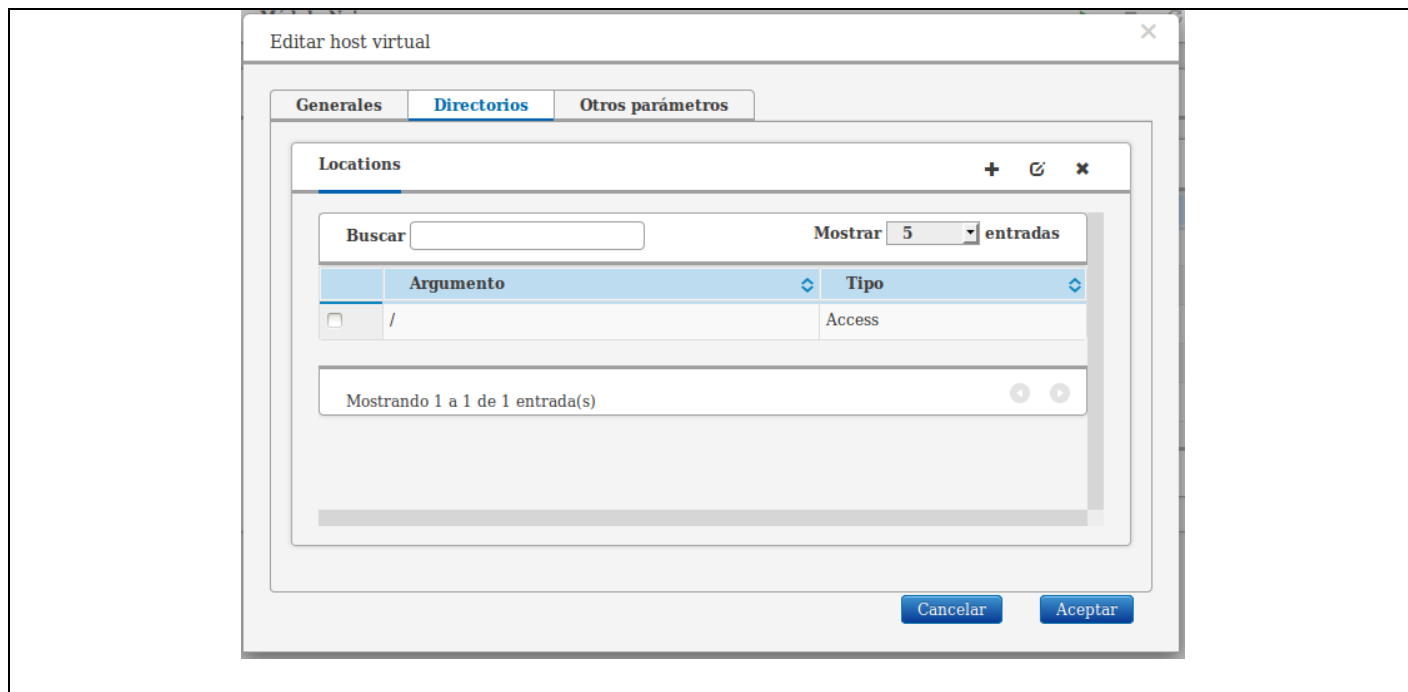


Tabla 26: HU Adicionar directorio.

<b>Número:</b> 17	<b>Nombre del requisito:</b> Adicionar directorio.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 20 horas	
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 15 horas	
<p><b>Descripción:</b> el sistema debe permitir al usuario adicionar los parámetros de alias y permisos de acceso pertenecientes a la directiva location. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente puede presionar el botón Adicionar o el Editar, en ambas interfaces existe una pestaña destinada a la gestión de los directorios, donde se muestra la lista de los mismos o un mensaje notificando si no existe ninguno. Se presiona el botón Adicionar para especificar los parámetros correspondientes.</p> <ul style="list-style-type: none"> <li>• Los permisos de acceso deben contener la raíz en el argumento de la directiva <i>location</i>. Se puede permitir o denegar el acceso al host virtual de todos los clientes web, o a una/varias direcciones IP</li> </ul>		

especificadas dentro de la directiva. Por defecto se permite el acceso al host virtual desde cualquier dirección IP del cliente.

- El alias debe contener en el argumento de la directiva *location* el nombre del alias que se desea poner y dentro de la directiva la ruta del host virtual con el nombre del alias.

**Observaciones:**

**Prototipo elemental de la interfaz gráfica de usuario:**

Tabla 27: HU Editar directorio.

<p><b>Número:</b> 18</p>	<p><b>Nombre del requisito:</b> Editar directorio.</p>
--------------------------	--

<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 20 horas
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 15 horas
<p><b>Descripción:</b> el sistema debe permitir al usuario editar los parámetros de alias y permisos de acceso pertenecientes a la directiva <i>location</i>. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente puede presionar el botón Adicionar o el Editar, en ambas interfaces existe una pestaña destinada a la gestión de los directorios, donde se muestra la lista de los mismos o un mensaje notificando si no existe ninguno. Debe existir al menos un directorio en la lista. Se presiona el botón Editar para especificar los parámetros correspondientes.</p> <ul style="list-style-type: none"> <li>• Los permisos de acceso deben contener la raíz en el argumento de la directiva <i>location</i>. Se puede permitir o denegar el acceso al host virtual de todos los clientes web, o a una/varias direcciones IP especificadas dentro de la directiva. Por defecto se permite el acceso al host virtual desde cualquier dirección IP del cliente.</li> <li>• El alias debe contener en el argumento de la directiva <i>location</i> el nombre del alias que se desea poner y dentro de la directiva la ruta del host virtual con el nombre del alias.</li> </ul>	
<b>Observaciones:</b>	
<b>Prototipo elemental de la interfaz gráfica de usuario:</b>	

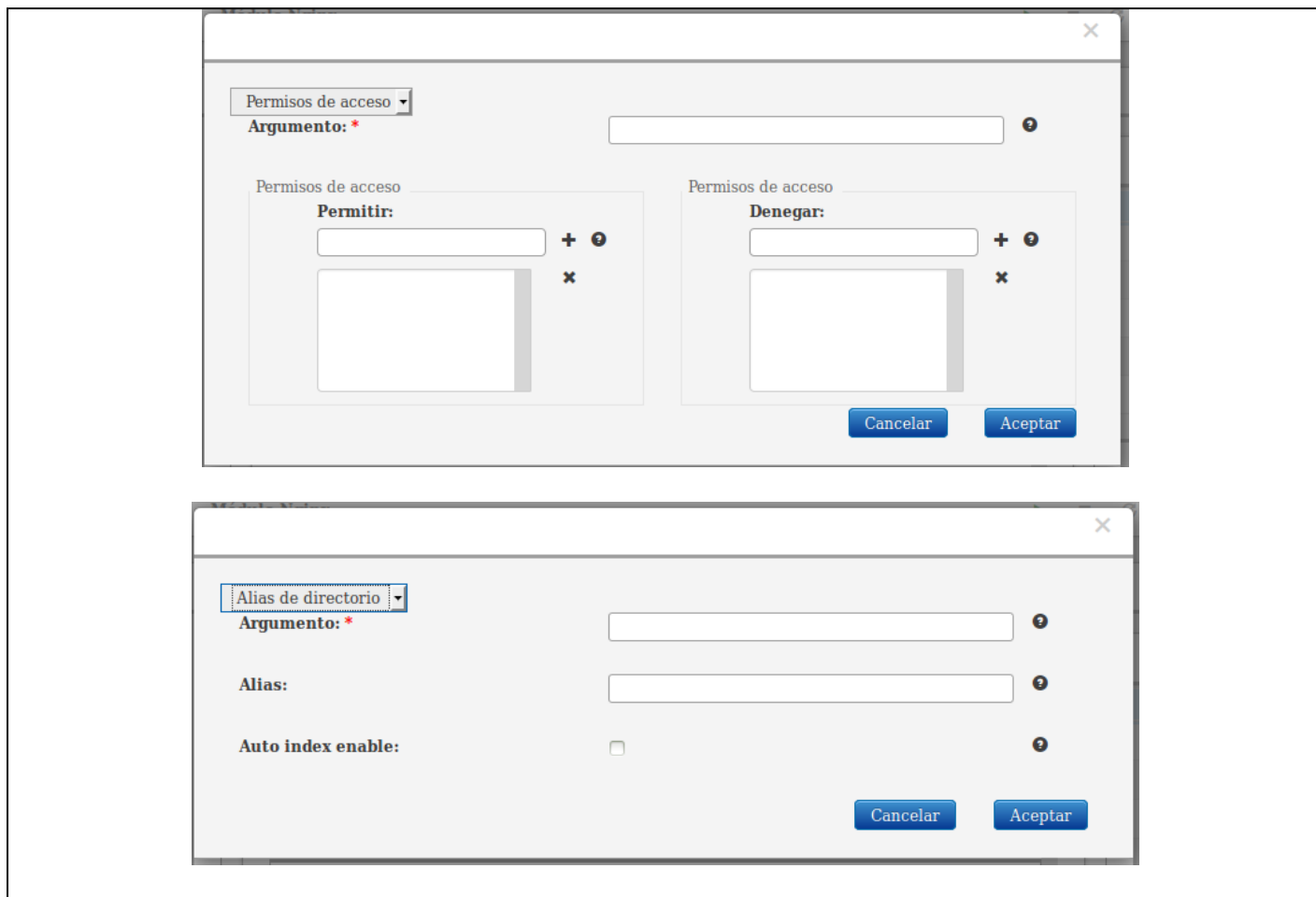


Tabla 28: HU Eliminar directorio.

<b>Número:</b> 20	<b>Nombre del requisito:</b> Eliminar directorio.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> El sistema debe permitir al usuario eliminar los parámetros de alias y permisos de acceso pertenecientes a la directiva <i>location</i> . La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente puede presionar el botón Adicionar o el		

Editar, en ambas interfaces existe una pestaña destinada a la gestión de los directorios, donde se muestra la lista de los mismos o un mensaje notificando si no existe ninguno. Debe existir al menos un directorio en la lista. El usuario selecciona el/los directorios deseados y presiona el botón Eliminar, se muestra un mensaje de confirmación, en caso positivo se ejecuta la eliminación y se notifica al usuario en caso satisfactorio o de error.

**Observaciones:**

**Prototipo elemental de la interfaz gráfica de usuario:**

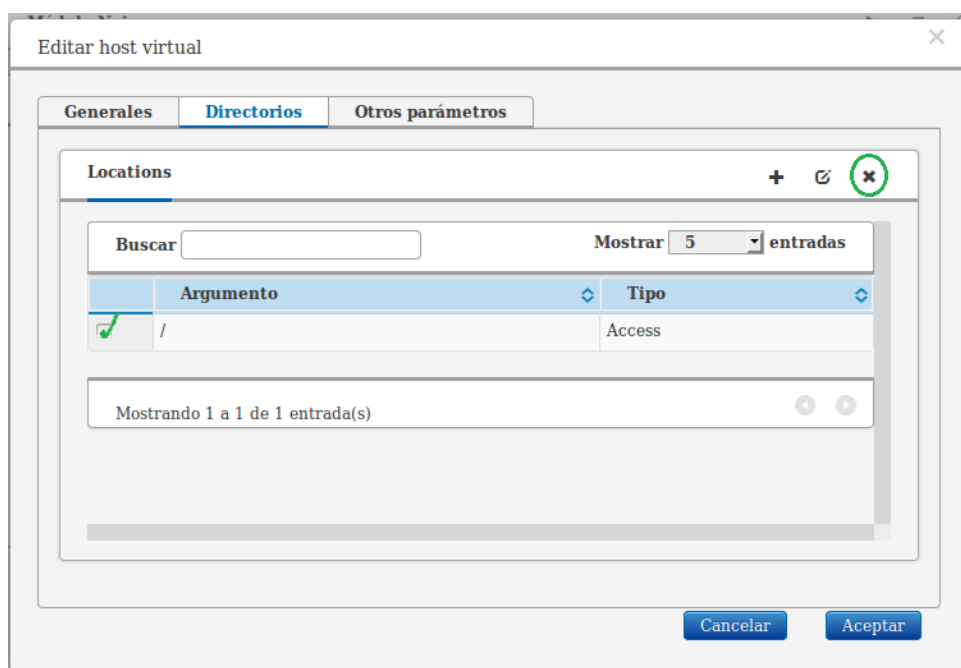


Tabla 29: HU Mostrar directorio.

<b>Número:</b> 19	<b>Nombre del requisito:</b> Mostrar directorio.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 10 horas	


<p><b>Descripción:</b> el sistema debe permitir al usuario mostrar los parámetros de alias y permisos de acceso pertenecientes a la directiva location. La funcionalidad comienza cuando el usuario accede al sistema y selecciona en el menú la opción Hosts Virtuales. Posteriormente puede presionar el botón Adicionar o el Editar, en ambas interfaces existe una pestaña destinada a la gestión de los directorios, donde se muestra la lista de los mismos o un mensaje notificando si no existe ninguno. Se presiona el botón Editar y en dicha interfaz se muestran los datos de un directorio.</p>
<p><b>Observaciones:</b></p>
<p><b>Prototipo elemental de la interfaz gráfica de usuario:</b></p> 

Tabla 30: HU Aplicar cambios locales.

<b>Número:</b> 23	<b>Nombre del requisito:</b> Aplicar cambios al servidor.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 10 horas	
<p><b>Descripción:</b> permite guardar las configuraciones locales del servicio en el servidor. El administrador del sistema o del módulo debe seleccionar la opción Aplicar cambios. Al ejecutar la opción se verifica que las configuraciones del servidor no hayan sufrido cambios.</p> <p>Si las configuraciones del servidor no han sufrido cambios se le muestra al usuario el mensaje: ¿Está seguro que desea salvar toda la configuración en el servidor? Esto reconfigurará el servicio con los datos establecidos. Luego el usuario debe seleccionar la opción Aceptar para guardar los cambios o la opción Cancelar para mantener la configuración anterior. Si selecciona la opción Aceptar, se muestra el mensaje: Los cambios han sido salvados.</p>		

Si las configuraciones del servidor han sufrido cambios se le muestra al usuario el mensaje: Por alguna razón uno o más archivos de configuración han sido modificados en el servidor. ¿Qué desea hacer?

- Aplicar los cambios: se salvarán los nuevos cambios realizados para el servidor, tenga en cuenta que con esta acción se perderán las modificaciones realizadas en el servidor.
- Descartar los cambios: se descartarían los cambios realizados y se restablecerán las configuraciones existentes en el servidor, tenga en cuenta que con esta acción se perderán todos los cambios no aplicados.
- Aplicar última configuración válida: se salvará la última configuración que existió en el servidor, tenga en cuenta que con esta acción no perderá los cambios sin aplicar, pero sí las modificaciones realizadas en el servidor.

El usuario debe seleccionar una de las tres opciones anteriores y luego seleccionar la opción Aceptar para guardar los cambios o la opción Cancelar para mantener la configuración anterior.

**Observaciones:** Tiene que existir algún cambio en las configuraciones locales del servidor para que esta funcionalidad se active.

**Prototipo elemental de la interfaz gráfica de usuario:**



Tabla 31: HU Descartar cambios locales.

<b>Número:</b> 24	<b>Nombre del requisito:</b> Descartar cambios locales.	
<b>Programador:</b> Rachel Molina Rodríguez	<b>Iteración Asignada:</b> 1	
<b>Prioridad:</b> Baja	<b>Tiempo Estimado:</b> 10 horas	
<b>Riesgo en desarrollo:</b> Baja	<b>Tiempo Real:</b> 10 horas	
<b>Descripción:</b> la funcionalidad comienza cuando el usuario selecciona la opción Descartar. Permite		



descartar las configuraciones locales, cambiando estas por las configuraciones del servidor. Luego al usuario se le muestra el mensaje: ¿Está seguro que desea descartar los cambios realizados hasta el momento? Esto causará pérdida en los cambios realizados. Posteriormente el usuario debe seleccionar la opción Aceptar para guardar los cambios o la opción Cancelar para mantener la configuración anterior.

**Observaciones:**

**Prototipo elemental de la interfaz gráfica de usuario:**



## Anexo 3: Acta de aceptación del producto.

**UCI** Universidad de las Ciencias Informáticas

## Acta de aceptación de productos de trabajo

### ACTA DE ACEPTACIÓN DE PRODUCTOS DE TRABAJO

En cumplimiento del **Convenio de colaboración** establecido entre el **Centro de Software Libre (CESOL)** y la estudiante **Rachel Molina Rodríguez** de la Facultad 1 de la Universidad de las Ciencias Informáticas y en función de la ejecución del proyecto: **Módulo Nginx de HMAST**, se hace entrega de los productos que se relacionan a continuación:

- Módulo Nginx de HMAST

La parte Cliente, luego de haber revisado los productos de trabajos relacionados anteriormente procede a firmar la aceptación de los mismos en total conformidad.

Entrega	Recibe
<b>Nombre y apellidos:</b> Rachel Molina Rodríguez	<b>Nombre y apellidos:</b> Yoandy Pérez Villazón
<b>Cargo:</b> Estudiante Facultad 1	<b>Cargo:</b> Jefe de CESOL
<b>Firma:</b>	<b>Firma:</b>

Fecha: 31/05/2017

Figura 9: Acta de aceptación del módulo Nginx de HMAST (elaboración propia).

**CamelCase:** es la práctica de escribir frases o palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra.

**FastCGI:** es un protocolo para interconectar programas interactivos con un servidor web. Es una variación de Interfaz Común de Entrada (CGI, del inglés *Common Gateway Interface*).

**GNU/Linux:** sistema operativo, más conocido como Linux.

**Hosts virtuales:** se refiere a que en un mismo servidor web se pueden hospedar múltiples proyectos cada uno con su propio dominio, aunque todos pertenezcan a la misma dirección IP pública.

**Mockup:** fotomontajes que permiten a los diseñadores gráficos y web mostrar al cliente la propuesta de sus diseños.

**Netcraft:** es una compañía de servicios de internet que ofrece análisis de cuota de mercado de servidores y alojamiento web.

**PC servidora:** computadora con un software servidor.

**Servidor web:** o servidor HTTP es un programa informático que procesa cualquier aplicación del lado del servidor.

**Sites-available:** directorio que contiene los sitios disponibles en Nginx.

**Sites-enabled:** directorio que contiene los sitios habilitados en Nginx.

**TCP/UDP:** protocolos de la capa de transporte.

**Wireframe:** esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web.