

Universidad de las Ciencias Informáticas

Facultad 1



**Título:** Mejora en el rendimiento del servidor Apache en el módulo  
Web de HMAST.

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

**Autor:**

Nestor Ariel Delgado Pacheco

**Tutores:**

Ing. Susana Sánchez Ortiz

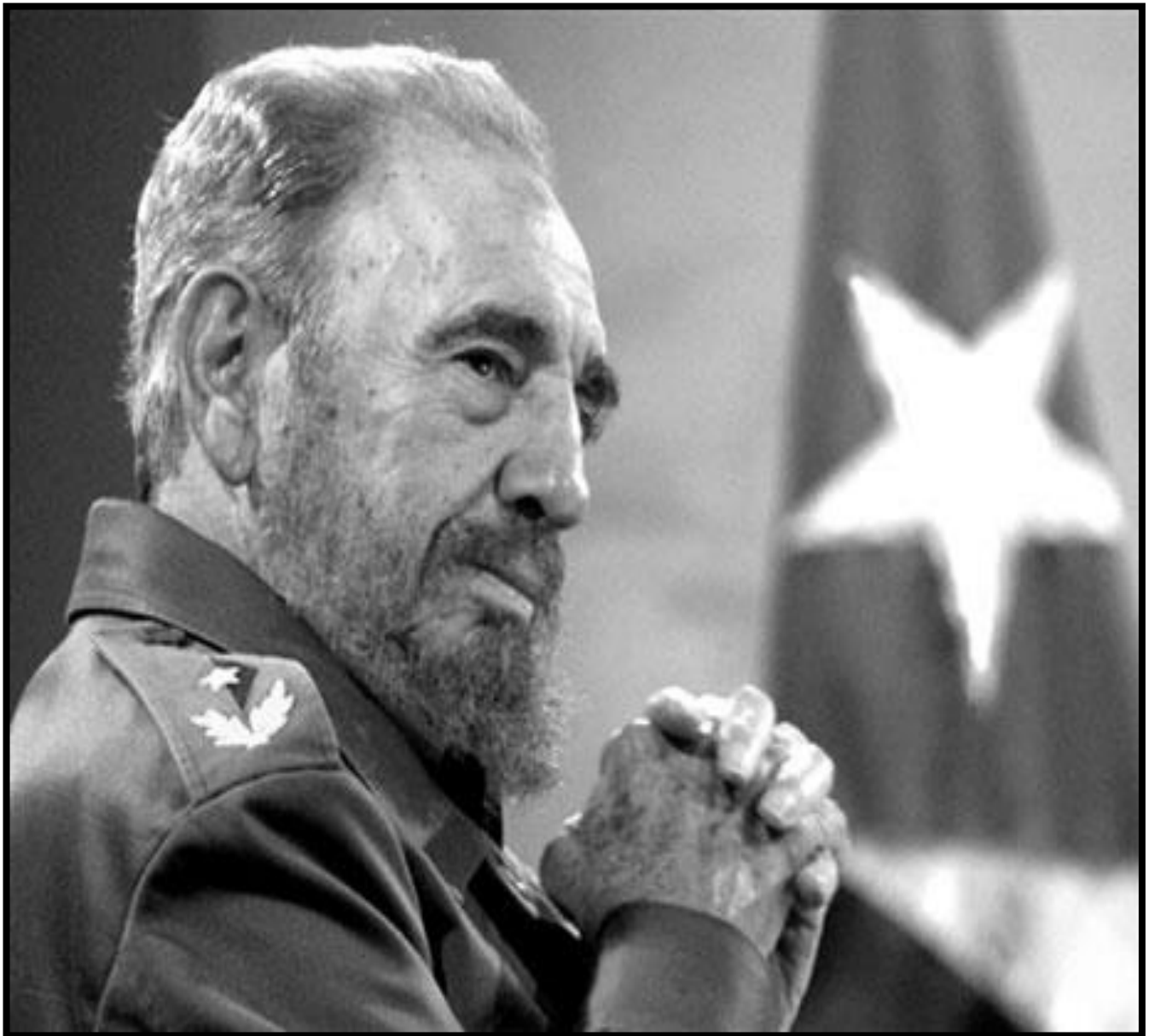
Ing. Alexander Pérez Tasé

**Consultante:**

Ing. Nadia Porro Lugo

**La Habana**

**Junio 2014**



*“La informática se convertirá en una poderosa fuerza científica, económica y política para Cuba”*

*Fidel Castro Ruz.*

## DECLARACIÓN DE AUTORÍA

---

Declaración de autoría

Declaro ser el único autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

**Nestor Ariel Delgado Pacheco**

Firma del Autor

**Ing. Susana Sánchez Ortiz**

Firma del Tutor

**Ing. Alexander Pérez Tasé**

Firma del Tutor

**Agradecimientos:**

A la Revolución: porque sin ella no hubiera sido posible este sueño.

A Fidel: por crear esta maravillosa Universidad.

A mis abuelos en especial a mi abuela Mirtha y a mi abuelo Sergio: por estar ahí en todo momento,  
por todo el amor y cariño dado durante todos estos años.

A mi abuelo Rene: que no pudo ver este sueño hecho realidad, gracias por todo lo que me enseñaste.

A mis padres Karelia y Juan Carlos: por apoyarme y acompañarme en cada decisión, por ayudarme en  
el cumplimiento de este sueño, por brindarme todo su amor y cariño, por ser mi guía y mi fortaleza  
en los momentos más difíciles.

A mi padrastro Cesar: por quererme como un hijo, por los múltiples consejos y enseñanzas, por  
educarme y brindarme todo su amor.

A mi tío Ulises: por ser tío y padre, gracias por tus consejos y enseñanzas.

A mi primo David: por estar siempre ahí, por soportarme y apoyarme.

A toda mi familia: por confiar en mí y tenerme presente.

A mis tutores Susana y Alexander: por guiarme certeramente en la realización y presentación de este  
trabajo.

A mis eternos amigos Osbel, Juan Pablo, Alejandro, Carlos y Reinier: gracias por su fiel amistad.

A mis hermanos de la UCI Jesús, Gustavo y Dany: por aguantarme todos estos años.

A mis compañeros de aula y apto: gracias por permitirme compartir con ustedes todos estos años.

A la FEU: por enseñarme día tras día a ser una mejor persona, por inculcarme principios y valores,  
por contribuir a formarme como persona y como profesional.

Al piquete de la FEU: esos que noche tras noche no descansan por hacer de la Facultad y la  
Universidad cada día la mejor, gracias por todas las enseñanzas, por siempre estar ahí, por su amistad,  
por ser mi segunda familia, fue todo un honor compartir con ustedes.

A todos los que directa e indirectamente ayudaron en mi crecimiento como persona y como  
profesional, especialmente a mi decano Damián, a Cecilia, Javier y Aylin.

A todos Muchas Gracias...

## Dedicatoria

Quiero dedicar el presente trabajo de diploma a:

Mis abuelos Mirtha y Sergio: por siempre confiar en mí, por brindarme todo su amor y cariño, por educarme, formarme y apoyarme en todo momento.

A mi mamá Karelia: lo más grande que tengo en este mundo, por todo el amor incondicional que me ha dado, por su apoyo y confianza, por creer y confiar en mí, gracias por todo mami.

A mi papá Juan Carlos: por transmitirme amor y cariño, por estar presente en los momentos buenos y malos. Gracias papá.

A mi padrastro Cesar: por ser mi guía, mi modelo a seguir, por el amor y cariño y los sabios consejos brindados. Gracias Padre.

## Resumen

El Departamento de Servicios Integrales en Migración Asesoría y Soporte perteneciente al Centro de Software Libre de la Universidad de las Ciencias Informáticas desarrolla la Herramienta de Migración y Administración de Servicios Telemáticos. Este sistema cuenta con diversos módulos implementados dentro de los que se encuentra el módulo Web, el cual permite la administración del servidor Web Apache2. El módulo Web presenta tres secciones de configuración: entorno global, servidor principal y *hosts* virtual; en la sección de entorno global se permite la configuración de los Módulos de Multiprocesamiento. El módulo Web configura solamente el Módulo de Multiprocesamiento *prefork* lo que impide que el servidor web se adapte a diferentes requerimientos tales como escalabilidad, disponibilidad y estabilidad. Estos elementos son necesarios para lograr un alto rendimiento de los recursos utilizados por el servidor Apache2. El objetivo de esta investigación se centra en la gestión de los Módulos de Multiprocesamiento y configuración de directivas globales del servidor web Apache2, que permita aumentar su rendimiento desde la Herramienta de Migración y Administración de Servicios Telemáticos. Para cumplir el objetivo planteado se realizó un estudio del estado del arte enfocado en el servidor web Apache2 y en los Módulos de Multiprocesamiento. Como resultado se implementaron los MPM *worker*, *event* y *threaded* y se realizó posteriormente un análisis del rendimiento del servidor web Apache2. El análisis realizado permitió comprobar el aumento del rendimiento del servidor web Apache2 desde el módulo Web, lo que facilita su utilización en las empresas e instituciones cubanas donde existe poco ancho de banda y condiciones mínimas en cuanto a *hardware*.

**Palabras claves:** Apache2, HMAST, Migración, Módulos, Multiprocesamiento

<b>INTRODUCCIÓN .....</b>	<b>9</b>
<b>CAPÍTULO 1: “EL SERVIDOR WEB APACHE2” .....</b>	<b>12</b>
1.1 SERVIDOR WEB APACHE2 .....	12
1.1.1 <i>Administración de Apache2</i> .....	13
1.1.2 <i>Funcionamiento de Apache2</i> .....	14
1.2 RENDIMIENTO EN APACHE2.....	15
1.3 LOS MPM EN APACHE2 .....	16
1.4 DESCRIPCIÓN DE LA HERRAMIENTA HMAST.....	19
1.4.1 <i>Arquitectura de HMAST</i> .....	19
1.4.2 <i>Funcionalidades que ofrece</i> .....	20
1.4.3 <i>Consideraciones para implementar un módulo para HMAST</i> .....	20
1.5 MÓDULO WEB DE HMAST .....	20
1.6 HERRAMIENTAS Y TECNOLOGÍAS A EMPLEAR .....	21
1.6.1 <i>Lenguaje de Programación</i> .....	21
1.6.2 <i>Entorno de Desarrollo Integrado (IDE)</i> .....	22
1.6.3 <i>Framework</i> .....	23
1.6.4 <i>Herramientas Case</i> .....	23
1.6.5 <i>Sistema de Control de Versiones</i> .....	24
1.6.6 <i>Herramienta de diseño gráfico Pencil</i> .....	24
1.6.7 <i>Augeas</i> .....	24
1.7 METODOLOGÍA DE DESARROLLO DE SOFTWARE .....	24
1.7.1 <i>Metodología SXP</i> .....	25
CONCLUSIONES PARCIALES .....	26
<b>CAPÍTULO 2: “ANÁLISIS Y DISEÑO DE LAS NUEVAS FUNCIONALIDADES DEL MÓDULO WEB PARA HMAST” .....</b>	<b>27</b>
2.1 PROPUESTA DE NUEVAS FUNCIONALIDADES DEL MÓDULO WEB PARA HMAST .....	27
2.2 ARTEFACTOS GENERADOS EN EL DESARROLLO .....	29
2.2.1 <i>Requisitos del Sistema</i> .....	30
2.2.2 <i>Historias de Usuario</i> .....	33
2.3 ARQUITECTURA DEL MÓDULO.....	41
2.4 DIAGRAMA DE PAQUETES .....	41
2.5 PATRONES DE DISEÑO EMPLEADOS .....	46
CONCLUSIONES PARCIALES .....	47
<b>CAPÍTULO 3: “IMPLEMENTACIÓN Y PRUEBAS DE LAS NUEVAS FUNCIONALIDADES DEL MÓDULO WEB PARA HMAST” .....</b>	<b>48</b>
3.1 PLANIFICACIÓN DE LA IMPLEMENTACIÓN.....	48
3.2 TAREAS DE INGENIERÍA .....	48
3.3 ESTÁNDARES DE CODIFICACIÓN.....	52
3.4 PRUEBAS AL MÓDULO WEB .....	53
3.4.1 <i>Pruebas de Aceptación</i> .....	53
3.5 ANÁLISIS DEL RENDIMIENTO .....	56
CONCLUSIONES PARCIALES .....	58
<b>CONCLUSIONES GENERALES .....</b>	<b>59</b>
<b>RECOMENDACIONES.....</b>	<b>60</b>

REFERENCIAS BIBLIOGRÁFICAS .....	61
ANEXOS .....	64
GLOSARIO DE TÉRMINOS .....	90



## Introducción

En Cuba ha alcanzado gran importancia el uso social y colectivo de las Tecnologías de la Información y las Comunicaciones (TIC), este proceso ha ido encaminado y enfocado a los pilares fundamentales del proceso revolucionario: la educación, la salud y la cultura. No se puede lograr una utilización masiva de las TIC sin haber elevado la calidad de la educación, instrumentando un proceso de educación continua y ampliando la cultura general de la población sobre el uso estas tecnologías.

Como dijo el compañero Fidel Castro en su discurso pronunciado en el Palacio Central de Computación por el 15 aniversario de este centro: *“La informática se convertirá en una poderosa fuerza científica, económica y política para Cuba”* [1]; por lo que es necesario asegurar la soberanía tecnológica, impulsar la innovación nacional y facilitar la inclusión digital. Actualmente el país se encuentra inmerso en un proceso paulatino de migración total hacia Software Libre y aplicaciones de código abierto, lo que constituye una necesidad en la sociedad cubana en los aspectos políticos, económicos, sociales y tecnológicos.

Con el propósito de apoyar la migración en el país, en la Universidad de las Ciencias Informáticas (UCI), fue creado el Centro de Software Libre (CESOL), el cual cuenta con el Departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS). La misión del mismo va dirigida a tres líneas fundamentales: el desarrollo de estrategias, guías y planes de migración; las herramientas de soporte a este proceso y la migración de los servicios telemáticos y aplicaciones de escritorio.

Como apoyo a este proceso se desarrolla la Plataforma de Migración [2] y la Herramienta de Migración y Administración de Servicios Telemáticos (HMAST) [3]. El desarrollo de HMAST tiene el propósito de disponer de una aplicación que se emplee en las diferentes empresas e instituciones cubanas que brinden los servicios telemáticos sobre sistemas operativos privativos. La misma está compuesta por un sistema base que tiene las funcionalidades necesarias para administrar los usuarios, las conexiones, las tareas programadas y los módulos. Por tanto, esta herramienta permitirá administrar y migrar los servicios telemáticos a plataformas libres.

Esta herramienta se encuentra en su primera versión y cuenta con varios módulos dentro de los que se encuentran: DHCP, SSH, Correo, Proxy, FTP y Web. El módulo Web permite la administración de las tres secciones de Apache2: entorno global, servidor principal y hosts virtuales en diferentes PC<sup>1</sup> que funcionan como servidores de forma remota.

Sin embargo, estas funcionalidades solo pueden ser consultadas o ejecutadas a nivel de consola, lo que

---

<sup>1</sup> Computadora Personal.

dificulta la utilización del mismo por los usuarios finales, pues se deben poseer conocimientos avanzados en la administración del servidor Apache2.

En Apache cada solicitud es atendida por un hilo separado o proceso y utiliza *sockets* sincrónicos. Con la creación de la versión 2.0 se introducen los Módulos de Multiprocesamiento (MPM). Solución propuesta por el Grupo Apache debido a que la anterior arquitectura no tenía un trabajo óptimo [4] bajo plataformas que no estaban centradas en procesos como en el caso del Sistema Operativo Windows. Los MPM permiten a los administradores del sistema ajustar Apache2 para ser más rápido según las necesidades y la naturaleza de las peticiones que tenga que atender.

El módulo Web desarrollado configura el MPM *prefork*, sin embargo no permite que se puedan configurar los restantes MPM, lo que impide que el servidor web se adapte a diferentes requerimientos tales como escalabilidad, disponibilidad y estabilidad.

Estos elementos son necesarios para lograr un alto rendimiento de los recursos utilizados por el servidor Apache2, lo que constituye una prioridad teniendo en cuenta el poco ancho de banda y las condiciones en cuanto a hardware que poseen las empresas e instituciones cubanas.

Para dar solución a la problemática anteriormente planteada se define como **problema científico** de la investigación: ¿Cómo proveer a la herramienta HMAST la capacidad de aumentar el rendimiento del servidor Apache2 desde el módulo Web?

Con vista a la solución del problema planteado el **objeto de estudio** del trabajo es el servidor web Apache2. Quedando enmarcado el **campo de acción** en la administración de Apache2 desde HMAST. Por lo que se establece como **objetivo general** desarrollar la gestión de los Módulos de Multiprocesamiento y configuración de directivas globales del servidor Apache2 que permita aumentar su rendimiento desde la herramienta HMAST. Con el propósito de lograr el objetivo propuesto se han definido los siguientes **objetivos específicos**:

- Caracterizar el servidor web Apache2 centrándose en los Módulos de Multiprocesamiento y en la configuración de sus directivas globales.
- Analizar y diseñar las funcionalidades propuestas.
- Implementar la capa de presentación del módulo Web y las funcionalidades propuestas.
- Probar las funcionalidades implementadas en el módulo Web.

Para lograr los objetivos propuestos se precisan las siguientes **tareas investigativas**:

- Caracterización del servidor web Apache2 a partir de la bibliografía especializada en el mismo.

- Caracterización de los Módulos de Multiprocesamiento y de la configuración de sus directivas globales.
- Análisis y diseño de las funcionalidades propuestas.
- Implementación de la capa de Presentación del módulo Web y las funcionalidades propuestas.
- Realización de las pruebas correspondientes a las funcionalidades implementadas.

El presente trabajo de diploma plantea la siguiente **idea a defender**: el desarrollo de la gestión de los Módulos de Multiprocesamiento y configuración de directivas globales del servidor Apache2 permitirá aumentar su rendimiento desde la herramienta HMAST.

Para el desarrollo de la investigación se utilizó el método científico **Analítico-Sintético** pues permitió el estudio de diferentes fuentes bibliográficas para poder extraer los elementos más importantes relacionados con el servidor web Apache2 y específicamente sus MPM. Se realizaron además resúmenes y valoraciones de conceptos relevantes relacionados con el tema.

La investigación está desarrollada en 3 capítulos, cuya estructura se describe a continuación:

En el primer capítulo, “**El servidor web Apache2**”, se realiza el estudio del estado del arte donde se hace referencia a los aspectos relacionados con el servidor Web Apache2 y con los Módulos de Multiprocesamiento, conceptos, características, funcionalidades e importancia. Se detallan las características del módulo Web para HMAST y se seleccionan además las tecnologías y la metodología de desarrollo de software orientada al desarrollo del sistema propuesto.

En el segundo capítulo, “**Análisis y diseño de las nuevas funcionalidades del módulo Web para HMAST**”, se hace referencia a la solución que se propone, definiendo la arquitectura a tener en cuenta durante el desarrollo de la misma. Además se describe el proceso ágil basado en las Historias de Usuarios correspondientes a las nuevas funcionalidades y prototipos de interfaz de usuario.

En el tercer capítulo, “**Implementación y pruebas de las nuevas funcionalidades del módulo Web para HMAST**”, se detallan aspectos relevantes de la implementación de los Módulos de Multiprocesamiento y de los aspectos propuestos tomando como guía las Tareas de Ingeniería generadas por la metodología seleccionada en el Capítulo 1. Se describen y se diseñan los casos de prueba y se plasman los resultados de su aplicación en el sistema.

## Capítulo 1: “El servidor web Apache2”

En el marco del estudio del estado del arte, en el presente capítulo se hará referencia a conceptos, características y funcionamiento del servidor web Apache2, específicamente de sus Módulos de Multiprocesamiento, así como la descripción del módulo Web de HMAST.

### 1.1 Servidor Web Apache2

Apache2 surge en 1995, es un servidor web de código abierto continuamente actualizado, adaptado a los nuevos protocolos, y cuya implementación se realiza de forma colaborativa. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Grupo Apache. Además de este grupo, cientos de personas han contribuido al proyecto con códigos, ideas y documentación [5]. Cuenta con distintas características dentro de las que se encuentran:

**Multiplataforma:** Es utilizado en multitud de sistemas operativos, lo que lo hace prácticamente universal. Actualmente funciona en Windows 2000/NT/9x, Linux, Unix, FreeBSD, Solaris, Mac OS X, entre otros.

**Sistemas de módulos dinámicos:** Se caracteriza por poseer un sistema de módulos dinámicos, los cuales pueden ser habilitados o deshabilitados mediante la ejecución de comandos o mediante el reinicio del servidor, lo que posibilita ser adaptado a diferentes entornos y necesidades lo cual es una ventaja pues permite que estén en marcha aquellas funcionalidades realmente necesarias, optimizando así el rendimiento.

**Soporte de *hosts* virtuales:** Es además uno de los primeros servidores web en soportar *hosts* virtuales basados en IP y/o basados en nombre. Puede proveer acceso a alojar cientos de sitios web clientes desde un único servidor.

**Introducción de Módulos de Multiprocesamiento en el manejo de peticiones:** Los MPM son responsables de conectar con los puertos de red de la máquina, aceptar las peticiones y generar los procesos hijos que se encargan de servirlos [6].

**Soporte del último protocolo HTTP 1.1:** Es uno de los primeros servidores Web en integrar el protocolo HTTP 1.1. Es totalmente compatible con el nuevo estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0. Apache2 está preparado para todas las novedades del nuevo protocolo. Por ejemplo, antes de HTTP 1.1, un navegador web tenía que esperar una respuesta del servidor web antes de poder emitir otra petición. Con el surgimiento de HTTP 1.1, esto ha dejado de ser así. Un navegador web puede enviar solicitudes en paralelo, las cuales ahorran ancho de banda

dejando de transmitir las cabeceras HTTP en cada solicitud. De algún modo se está ofreciendo un estímulo del lado del usuario final porque los archivos solicitados en paralelo aparecerán antes en el navegador [7].

**Configuración simple y potente a través de ficheros:** No posee una interfaz de usuario gráfica para su administración. Su configuración se realiza mediante un archivo de configuración llamado `httpd.conf` o `apache2.conf`. Sin embargo permite repartir la configuración de su *host* virtual en múltiples archivos de configuración, permitiendo que su fichero principal no sea demasiado grande y complejo a la hora de configurar.

Apache2 presenta además otras características como [8]:

- Soporte para CGI (*Common Gateway Interface*).
- Soporte de FastCGI.
- Soporte de autenticación HTTP.
- Perl integrado.
- Soporte de *scripts* PHP.
- Soporte de *servlets* de Java.
- Servidor proxy integrado<sup>2</sup>.
- Soporte de SSI (*Server Side Includes*).
- Soporte de SSL (*Secured Socket Layer*).
- Filtrado I/O<sup>3</sup>.

### 1.1.1 Administración de Apache2

En la presente investigación se hace referencia a la administración de Apache2 mediante el archivo de configuración llamado `apache2.conf` o `httpd.conf` teniendo en cuenta el uso de los sistemas operativos Nova, Ubuntu o Debian.

Las configuraciones se pueden separar por partes en diversos ficheros. Un ejemplo es poder aislar la configuración de cada *host* virtual<sup>4</sup> en un archivo independiente para cada uno, o bien distribuir la configuración de diferentes localizaciones físicas a ficheros independientes más pequeños y fáciles de manejar que se encuentren directamente en dichas localizaciones. No es necesario concentrar toda la configuración en un solo archivo de gran tamaño. De esta forma se evita que el fichero de configuración principal `apache2.conf` alcance un tamaño demasiado grande y por tanto sea más complejo de manejar [11]. En el directorio `/etc/apache2` se encuentra toda la configuración referente a Apache2 el cual

---

<sup>2</sup> Servidor proxy integrado: Apache se puede convertir en un *caching (forward)* [9].

<sup>3</sup> Filtrado I/O: Otro cambio importante en Apache 2.0 es que proporciona arquitectura para I/O jerarquizada [10].

<sup>4</sup> Directiva de Apache2 que describe la configuración de los sitios web alojados en el servidor.

incluye:

- `apache2.conf`: Fichero principal de configuración de Apache2.
- `httpd.conf`: Fichero de configuración que se mantiene por compatibilidad con la anterior versión de Apache y por algunos módulos especiales. Se considera obsoleto.
- `mods-available`: Directorio con los módulos disponibles. Cada módulo consta de un fichero para cargar (`.load`) y otro, opcional, para la configuración (`.conf`). Los ficheros binarios de los módulos (`.so`) están en `/usr/lib/apache2/modules`.
- `mods-enabled`: Directorio con los módulos activos. Normalmente son enlaces simbólicos a los ficheros de un módulo de la carpeta anterior.
- `ports.conf`: Fichero para configurar el puerto de comunicación HTTP. Por defecto, Apache2 escucha por el puerto 80 y el 443 si SSL está activo.
- `README`: Documentación sobre este directorio.
- `sites-available`: Directorio con los ficheros de configuración de los sitios web disponibles. Se recomienda configurar cada sitio con su fichero.
- `sites-enabled`: Directorio con los ficheros de configuración de los sitios web activos. Se usa la misma filosofía que para activar módulos. Si se quiere activar un sitio se debe crear un enlace directo en esta carpeta que apunte a `sites-available`.

La configuración se realiza mediante directivas y secciones. Las directivas son variables almacenadas en el archivo de configuración para alterar y controlar el funcionamiento de Apache2 en tiempo de ejecución según sus valores y después de haber reiniciado el proceso en el servidor.

El archivo de configuración principal de Apache2 está dividido en tres secciones fundamentales, aunque las directivas de cada una pueden aparecer mezcladas y desordenadas.

- **Sección 1:** Entorno Global. Controla el comportamiento general de Apache2.
- **Sección 2:** Servidor Principal. Se describe la configuración general del servidor.
- **Sección 3:** *Hosts* Virtuales. Agrupa las directivas relacionadas con los *hosts* virtuales que se definan.

### 1.1.2 Funcionamiento de Apache2

Apache2 presenta por defecto un sitio web que es la base sobre el cual se construyen los *hosts* virtuales, el cual es llamado servidor principal. Apache2 soporta los *hosts* virtuales basados en nombres y los basados en IP.

Una PC servidora puede tener asignadas una o varias direcciones IP y cada dirección IP puede escuchar por una lista de puertos específicos y además por todos los puertos, que en la configuración

se representa con el símbolo de asterisco (\*). A cada combinación direcciónIP:puerto se puede asignar un *host* virtual basado en IP o una lista de *hosts* virtuales basados en nombre [12].

Un mismo *host* virtual puede estar asignado a varias combinaciones direcciónIP:puerto. Al iniciarse Apache2 se genera una lista por cada combinación direcciónIP:puerto y se inserta en una tabla *hash*. Cuando llega una petición al servidor se realiza una búsqueda del *host* virtual que debe atenderla. En caso de no existir coincidencias entre la dirección IP y/o el puerto de la petición con los asignados a los *hosts* virtuales, la petición será atendida por el servidor principal.

## 1.2 Rendimiento en Apache2

La palabra rendimiento hace referencia a la capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas [13]. Este término es muy usado en las ciencias de la informática principalmente por los administradores de servidores web. En un servidor web es la capacidad para tratar solicitudes HTTP y reducir los tiempos de respuestas de las aplicaciones, lo que mejora la experiencia de los clientes que tienen acceso a los mismos.

Apache2 es un servidor de propósito general, diseñado para proporcionar un equilibrio de flexibilidad, portabilidad y rendimiento. Apache2 presenta características que le permiten lograr una alta capacidad de respuestas en distintas situaciones. La mayoría de estas características están habilitadas de forma predeterminada. Sin embargo presenta opciones de configuración en tiempo de compilación y en tiempo de ejecución que pueden afectar significativamente el rendimiento, permitiendo un mejor aprovechamiento de las capacidades del *hardware* y del sistema operativo [14]. Las características más significativas que presenta Apache2 para lograr obtener un alto rendimiento son:

- **Diseño Modular:** Apache2 puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona. En Apache2 existen infinidad de módulos disponibles, todos ellos denominados `mod_XXX`, donde el nombre XXX designa el tipo de funcionalidad que está pensado que desempeñen. La existencia de estos dota a Apache2 de la capacidad de desempeñar prácticamente cualquier función que se pueda necesitar. Esta orientación permite al servidor ser personalizado de forma muy precisa, instalando o desinstalando servicios a demanda de los usuarios del mismo, y que de esta forma sólo estén en marcha aquellas funcionalidades realmente necesarias. Tiene como ventaja un menor consumo de recursos de manera general, al poder desactivar todo lo que no sea necesario y también disminuir la probabilidad de ataques, al reducir la cantidad de servicios en marcha que se deben mantener. No obstante, solo debe instalarse aquello que realmente vaya a ser usado. Si se hace

un mal uso de la capacidad de modularización, se pierden todas sus ventajas y se pueden ocasionar múltiples problemas [15].

- **Configuración en Tiempo de Ejecución:** Apache2 permite la configuración de distintas directivas en tiempo de ejecución. Su configuración correcta permite elevar el rendimiento del servidor, adaptándolo eficientemente a las distintas situaciones. A continuación se desglosan las diferentes directivas:
  - *HostnameLookups:* Activa la resolución de DNS de las direcciones IP de los clientes.
  - *AllowOverrideAll:* Directiva que busca los ficheros de configuración.
  - *KeepAlive:* Permite que se establezcan conexiones HTTP persistentes.
- **Módulos de Multiprocesamiento:** Cuando se configura Apache2 se debe elegir un MPM a usar. Hay MPM específicos para algunas plataformas como: beos, mpm network, mpm os2 y mpm winnt, los cuales son explicados en el siguiente epígrafe. Para los sistemas de tipo Unix generales existen varios MPM. La adecuada selección de los mismos regula la velocidad y la escalabilidad del servidor Apache2 y su utilización permiten trabajar con un mayor rendimiento [16].

### 1.3 Los MPM en Apache2

Apache2 está diseñado para ser un servidor web que puede funcionar en la más amplia variedad de plataformas y entornos. Lo que hacen que a menudo sean necesarias diferentes características o funcionalidades o que una misma característica o funcionalidad sea implementada o configurada de diferentes maneras para obtener una mayor eficiencia. El diseño modular de Apache2 permite a los administradores de sitios web elegir qué funcionalidades van a ser incluidas en el servidor y seleccionar qué módulos se van a utilizar, ya sea al compilar o al ejecutar el servidor.

Apache2 extiende este diseño modular hasta las funcionalidades más básicas de un servidor web. El servidor viene con una serie de Módulos de Multiprocesamiento que son los responsables de conectar con los puertos de red de la máquina, aceptar las peticiones y generar los procesos hijos que se encargan de servirlos [17].

La extensión del diseño modular a este nivel del servidor ofrece dos beneficios importantes:

1. Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos. En concreto, la versión de Windows de Apache es mucho más eficiente, porque el módulo *mpm\_winnt* puede usar funcionalidades nativas de red en lugar de usar la capa POSIX<sup>5</sup>

---

<sup>5</sup> POSIX: estándar de interfaz de sistemas operativos



como hace Apache 1.3. Este beneficio se extiende también a otros sistemas operativos que implementan sus respectivos MPM.

2. El servidor puede personalizarse mejor para las necesidades de cada sitio web. Por ejemplo, los sitios web que necesitan más que nada escalabilidad pueden usar un MPM hebrado<sup>6</sup> como *worker*, mientras que los sitios web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar *prefork*. Además, se pueden configurar funcionalidades especiales como servir diferentes *hosts* con diferentes identificadores de usuario (*perchild*).

Para los usuarios los MPM son como cualquier otro módulo de Apache2. La diferencia más importante es que solo uno puede estar cargado en el servidor en un determinado momento. Los MPM de Apache2 son: *mpm\_common*, *mpmt\_os2*, *beos*, *mpm\_netware*, *mpm\_winnt*, *prefork*, *worker*, *threaded*, *event* y *perchild* [18]. A continuación se explicarán los más utilizados.

- **El MPM Prefork**

Este módulo crea un grupo de procesos hijos para servir peticiones. Cada proceso hijo tiene un sólo hilo, por lo que si Apache2 inicia 20 procesos hijos, puede servir 20 peticiones simultáneamente. En el caso de que ocurra un error grave y un proceso hijo muera, sólo se pierde una petición: la que está atendiendo el hijo que acaba de cesar la actividad. El máximo y el mínimo de procesos hijos son parámetros configurables, permitiendo adaptar Apache a las capacidades de la PC servidora y también fijarle un máximo de carga. Dentro de estos parámetros máximo y mínimo el servidor genera más o menos procesos en función de la carga de trabajo actual [19].

- **El MPM Worker**

Este MPM implementa un servidor híbrido multiproceso-multihebra. Usando hilos<sup>7</sup> para atender peticiones, el servidor puede servir un mayor número de peticiones con menos recursos del sistema que un servidor basado únicamente en procesos. Como Apache2 mantiene un mínimo de procesos hijos inactivos esperando peticiones, los clientes no esperan la creación de un nuevo proceso hijo para atenderle sino que el proceso hijo usa un hilo que se encarga de la petición. No obstante, se mantiene casi por completo la estabilidad de un servidor basado en procesos manteniendo la capacidad multiproceso, pudiendo cada proceso tener muchas hebras<sup>8</sup> [20].

---

<sup>6</sup> Hebrado: Término utilizado por la bibliografía de Apache2 para referirse a un servidor multiproceso.

<sup>7</sup> Hilos: Unidad de procesamiento más pequeña que puede ser planificada por un programa.

<sup>8</sup> Hebras: Varios hilos de ejecución.

- **El MPM *Threaded***

Es como el MPM *prefork*, pero en vez de que cada proceso hijo tenga un solo hilo, cada proceso hijo puede tener un número determinado de hilos. Si Apache2 inicia 30 procesos hijos en los que cada hijo puede tener un máximo de 10 hilos, entonces Apache2 podrá servir  $30 \times 10 = 300$  solicitudes simultáneas. Dado que los hilos son más eficaces en cuanto a recursos que los procesos, este MPM es muy escalable [21].

- **El MPM *Event***

Se basa en el MPM *worker*, intenta solucionar el problema HTTP de *KeepAlive*<sup>9</sup>. Cuando un cliente completa la primera petición, el cliente puede mantener abierta la conexión y enviar más peticiones usando el mismo *socket*. Esto puede mejorar el rendimiento en la creación de conexiones TCP, sin embargo, Apache2 tradicionalmente mantiene un hijo completo *process/thread* esperando los datos de un cliente lo que tiene sus propias desventajas. Para resolver este problema MPM *event* usa un único *thread* dedicado a manejar tanto los *sockets* de escucha como todos los *sockets* que están en un estado *KeepAlive* [22].

- **El MPM *Perchild***

Este MPM inicia un conjunto de procesos hijos con un número determinado de hilos. A medida que aumenta la carga de solicitudes, los procesos añaden nuevos hilos según los van necesitando. Cuando se reduce el conteo de solicitudes, los procesos disminuyen sus conteos de hilos utilizando un máximo y un mínimo asignado de conteo. Además este MPM agrega la capacidad extra para especificar que determinados procesos deben atender las solicitudes en virtud de otro usuario-IDs. Estos usuarios-IDs se asocian con los hosts virtuales específicos. Este MPM se encuentra no funcional [23].

Los módulos de multiprocesamiento que se van a usar posteriormente deben elegirse durante el proceso de configuración y deben ser compilados en el servidor. Los compiladores son capaces de optimizar muchas funciones si se usan hebras.

Si durante el proceso de configuración de Apache2 no se determina qué MPM se va a utilizar, Apache2 selecciona un MPM por defecto. En la siguiente tabla se muestran los MPM por defecto para varios sistemas operativos:

---

<sup>9</sup> KeepAlive: conexiones HTTP persistentes.

Sistema Operativo	MPM por defecto
BeOS	beos
Netware	mpm_netware
OS/2	mpmt_os2
Unix	prefork
Windows	mpm_winnt

Tabla1.1 MPM por defecto para cada Sistema Operativo

## 1.4 Descripción de la Herramienta HMAST

HMAST cuenta con un sistema base que permite la administración de las PC servidoras de forma remota, el cual tiene las funcionalidades necesarias para administrar los usuarios, las tareas programadas y los servicios.

### 1.4.1 Arquitectura de HMAST

La arquitectura presentada por la herramienta HMAST propone el diseño de una arquitectura N-Capa orientada al dominio y distribuida en cinco componentes o paquetes. La interacción entre los mismos se realiza a través de interfaces y utilizando inyección de dependencias.

La **capa de Presentación** es responsable de mostrar información al usuario e interpretar sus acciones. Sus componentes implementan la funcionalidad requerida para que los usuarios interactúen con la aplicación. La **capa de Aplicación** realiza las llamadas a servicios de la capa inferior y tiene la responsabilidad de adaptar la información que le llega, a los requerimientos de los servicios de dominio. La **capa de Dominio** constituye el hilo conductor de la aplicación, sus componentes solo dependen de la Capa de Infraestructura Transversal. Implementa la lógica de dominio (reglas de negocio) y es responsable de las validaciones. Define las interfaces de persistencia a datos (contratos de repositorio) pero no los implementa. Sus componentes no están ligados a tecnologías específicas. La **capa de Persistencia** es responsable de contener el código necesario para persistir los datos, contiene como componente los repositorios que son clases que implementan los contratos de repositorios definidos en la capa de dominio. Las responsabilidades de la **capa de Infraestructura Transversal** están dadas a promover la reutilización de código, contiene las operaciones de seguridad, *logging*, monitoreo del sistema, mecanismos de persistencia reutilizables, validadores genéricos y todas aquellas operaciones

que se puedan llamar desde otras capas [24].

### 1.4.2 Funcionalidades que ofrece

La herramienta cuenta con funcionalidades como son [25]:

- **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, entre otros).

### 1.4.3 Consideraciones para implementar un módulo para HMAST

Para la implementación de un módulo que se desee integrar en HMAST se debe tener en cuenta que:

- 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, llamadas a componentes de Infraestructura para que realicen tareas complementarias.
- Se debe garantizar que no se envíen hacia y desde la capa de Presentación objetos de Dominio, en su lugar deben viajar Objetos de Transferencia de Datos (DTO, *Data Object Transfer*).
- Las clases de servicios deben ser las únicas responsables (vías de acceso) 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.
- Es importante que todo el código reutilizable por más de un repositorio se ponga a disposición de todos en la capa de Infraestructura Transversal.

### 1.5 Módulo Web de HMAST

El módulo Web de HMAST permite la administración de los servidores Web en dicha herramienta, asegurando su adaptabilidad en las diferentes empresas e instituciones cubanas. El módulo Web administra Apache2, por lo que el sistema está estructurado basándose en su funcionamiento. El módulo permite la administración de Apache2 en diferentes PC servidoras de forma remota. Se puede administrar las tres secciones de configuración de Apache2 vistas anteriormente, donde la configuración del entorno global afecta tanto al servidor principal como a los hosts virtuales, ya sean basados en

nombres o basados en IP. Facilita un mejor entendimiento por parte de los usuarios para la gestión de los hosts virtuales debido a que se muestra de forma detallada la estructura de almacenamiento lógico de los mismos. El módulo Web permite realizar las siguientes funcionalidades de Apache2 [26]:

- **Gestionar hosts virtuales asignados a Apache2:** Permite al usuario mostrar, adicionar, eliminar, modificar, habilitar y deshabilitar hosts virtuales.
- **Permite Manejar los estados de Apache2:** Se manejan los estados del servidor Apache2 realizando diferentes acciones como: instalar, iniciar, reiniciar, detener y recargar.
- **Modificar puertos para las conexiones en Apache2:** Permite modificar los puertos para las conexiones en Apache2.
- **Especificar los módulos a usar en Apache2:** Permite al usuario mostrar, habilitar, y deshabilitar módulos.
- **Especificar parámetros en el servidor principal de Apache2:** Muestra y modifican algunos parámetros en el servidor principal de Apache2.
- **Especificar parámetros en el MPM *prefork* de Apache2:** Permite mostrar y modificar los parámetros de configuración pertenecientes al MPM *prefork*.
- **Especificar parámetros para el manejo de conexiones en Apache2:** Permite manejar el estado de las conexiones en Apache2.

El módulo Web se encuentra desarrollado bajo una arquitectura N-Capas, haciendo uso de la arquitectura definida para la herramienta HMAST, explicada en el epígrafe anterior, permitiendo así su integración con la misma. El desarrollo de este módulo en el año 2013 no incluyó la interfaz visual, por lo que carece de una capa de Presentación que facilite su uso por los usuarios finales. Las capas de Aplicación, Dominio y Persistencia tienen insertado un paquete con todo lo referente al módulo. El paquete correspondiente a la capa de Infraestructura Transversal contiene todas las clases que son reutilizables. Por lo que se hace necesario la selección de herramientas y tecnologías a emplear en el desarrollo de dichas capas.

## 1.6 Herramientas y Tecnologías a emplear

Para el desarrollo de las nuevas funcionalidades propuestas para el módulo Web se definió el uso de las siguientes herramientas y tecnologías, las cuales constituyen un requisito para poder integrar el módulo a la herramienta HMAST.

### 1.6.1 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 [27]. El lenguaje de programación seleccionado fue:

- **Java**

Java es un lenguaje de programación orientado a objetos, desarrollado por *Sun Microsystems* a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador. Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación. Permite abrir *sockets*, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red [28].

### 1.6.2 Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto [29]. El IDE a utilizar es:

- **Netbeans**

Netbeans es un Entorno de Desarrollo Integrado (IDE) gratuito de código abierto, hecho principalmente para el lenguaje de programación Java. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis, entre otros. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, aunque permite crear aplicaciones con Python, identificar errores y el debugger [30]. El *framework* a utilizar será:

### 1.6.3 Framework

Un *framework*<sup>10</sup> es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio [31].

- **Spring**

Spring *Framework*, también conocido simplemente como Spring, es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java, aunque también existe una versión para la plataforma .NET. Por su diseño, ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. Aunque contiene muchas características organizadas en grandes módulos, que le dan una funcionalidad muy amplia, sus características principales son la inyección de dependencias y la programación orientada a aspectos [27]. Spring fue seleccionado fundamentalmente por permitir la inyección de dependencias para lograr un bajo acoplamiento entre los objetos de la aplicación y la Programación Orientada a Aspectos (AOP), que es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización<sup>11</sup> de las aplicaciones y posibilitar una mejor separación de conceptos [32].

### 1.6.4 Herramientas Case

Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras [33].

- **Visual Paradigm para UML**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y

---

<sup>10</sup> Framework: Marco de Trabajo

<sup>11</sup> Modularización: Dividir un problema en segmentos simples

a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML [34].

### **1.6.5 Sistema de Control de Versiones**

Un sistema de control de versiones (o sistema de control de revisiones) es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web [35].

- **RapidSVN**

RapidSVN es un cliente de interfaz gráfica para la comunicación con servidores Subversion. Está escrito en C++ y distribuido bajo licencia GPL. Este facilita el versionado de ficheros, desde una interfaz sencilla e intuitiva y se encuentra disponible para plataformas Windows, Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente.

### **1.6.6 Herramienta de diseño gráfico Pencil**

Pencil es un software gratuito y de código abierto para diseñar gráficos y animarlos en dos dimensiones. La interfaz gráfica de Pencil dispone de las herramientas básicas para crear elementos gráficos, ya sea en formato de mapa de bits o vectorial, opciones de las herramientas tan sencillas como el tamaño, el color, una práctica paleta de colores y varias opciones de visualización.

### **1.6.7 Augeas**

Augeas es una herramienta de edición de ficheros de configuración del sistema GNU/Linux. Es un editor que analiza los archivos en sus formatos nativos y los transforma en un tipo abstracto de datos llamado árbol y viceversa. La manipulación de la configuración la realiza mediante este árbol y los cambios que sean aplicados a él se guardan de forma nativa en dichos archivos de configuración. Un nodo del árbol tiene asignada una etiqueta y un valor, además le corresponden varios hijos que son una lista de nodos que pueden tener la misma etiqueta. 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 [36].

## **1.7 Metodología de desarrollo de Software**

Las metodologías de desarrollo de software son el conjunto de procedimientos, técnicas, herramientas y



un soporte documental, que ayuda a los desarrolladores a realizar nuevo software. La metodología define quién debe hacer qué, cuándo y cómo debe hacerlo para obtener los distintos productos parciales y finales. Se clasifican en dos tipos: tradicionales y ágiles.

Las tradicionales o pesadas se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. No se adaptan adecuadamente a los cambios, ya que se aplican en entornos donde los requisitos están perfectamente bien definidos y estables.

Por otra parte, las metodologías ágiles están orientadas al resultado del producto y no a la documentación. Dan mayor valor al factor humano, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Se aplican en entornos donde los requisitos no pueden predecirse o bien pueden variar, permitiendo que el proceso sea adaptable y que se puedan realizar cambios de último momento [37].

Por las ventajas que ofrecen estas últimas, se propone para el desarrollo del trabajo de diploma el uso de las metodologías ágiles, específicamente SXP.

### **1.7.1 Metodología SXP**

SXP es un híbrido de metodologías ágiles que toma las mejores prácticas de las metodologías SCRUM y XP además de regirse por los lineamientos de calidad de la UCI. Con la aplicación de la metodología SXP se logra una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos. Teniendo como premisa una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

Consta de 4 fases principales, Planificación-Definición: se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; Desarrollo: se realiza la implementación del sistema hasta que esté listo para ser entregado. Entrega que es la responsable de la puesta en marcha y Mantenimiento: se realiza el soporte para el cliente.

Es una metodología que se caracteriza por ser iterativa e incremental con pequeñas mejoras unas tras otras, basada en Historias de Usuario (HU), está atenta al cambio y permite que el equipo de programación se mantenga en frecuente interacción con el cliente o usuario. Está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo [38].

### **Conclusiones Parciales**

Mediante la revisión de la bibliografía especializada en el servidor web Apache2, se ha logrado la comprensión de las características, funcionamiento y administración del mismo. La definición de las características y funcionamiento del módulo Web de HMAST el cual administra Apache2 permitió su caracterización para la posterior definición de las nuevas funcionalidades a implementar. El estudio de la herramienta HMAST permitió definir los requerimientos que debe seguir el módulo para integrarse a la herramienta. Las tecnologías a emplear para el desarrollo del módulo debido a que son las que se requieren en HMAST son: framework Spring, lenguaje de programación Java, IDE Netbeans, Visual Paradigm para UML como herramienta CASE, RapidSVN como sistema de control de versiones, Pencil para el diseño gráfico y Augeas para la configuración de ficheros; donde todo el proceso estará guiado por la metodología ágil de desarrollo de software SXP.

## Capítulo 2: “Análisis y diseño de las nuevas funcionalidades del módulo Web para HMAST”

Una vez analizadas las características y funcionalidades del módulo Web y detallados los requerimientos que se necesitan en HMAST, en el presente capítulo se definirán las nuevas características y funcionalidades que tendrá el módulo, las tareas a realizar para darle cumplimiento al desarrollo de las mismas, así como la arquitectura y patrones de diseño a emplear.

### 2.1 Propuesta de nuevas funcionalidades del módulo Web para HMAST

La presente investigación propone la implementación de nuevas funcionalidades al módulo Web para HMAST permitiendo realizar la administración y configuración de los MPM *worker*, *event* y *threaded* de forma que se pueda incrementar el rendimiento del servidor Apache2 desde el módulo Web. No se implementa el MPM *perchild* pues durante la revisión de la bibliografía especializada y como se describe en el Capítulo 1 Epígrafe 1.3 se encuentra no funcional. Además se permitirá la administración y configuración de directivas globales las cuales tienen como función principal optimizar el tiempo de respuesta del módulo Web. En adición a lo anterior se implementará la capa de Presentación permitiendo que el módulo Web cuente con una interfaz gráfica sencilla que permita al usuario final la administración y configuración de Apache2 de una forma más eficiente. A continuación se explican las principales características de las nuevas funcionalidades a implementar al módulo Web para HMAST.

- **El MPM Worker**

El MPM worker permitirá mostrar y modificar los parámetros de configuración pertenecientes al mismo. Al instalar `apache2-mpm-worker` desinstalará el módulo `php5` que no es compatible con ese módulo e instalará `php5-cgi`. Cuenta con las siguientes directivas de configuración:

- Número de procesos hijos que se crean al iniciar Apache (*StartServers*).
- Límite superior del número configurable de procesos (*MaxClients*).
- Número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*).
- Número máximo de hebras en espera (*MaxSpareThreads*).
- Número de hebras creadas por cada proceso hijo (*ThreadsPerChild*).
- Límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*).
- Límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*).

- **El MPM *Threaded***

El MPM *threaded* permitirá mostrar y modificar los parámetros de configuración pertenecientes al mismo. Cuenta con las siguientes directivas de configuración:

- Límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*).
- Límite superior del número configurable de procesos (*MaxClients*).
- Número máximo de hebras en espera (*MaxSpareThreads*).
- Número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*).
- Número de procesos hijos que se crean al iniciar Apache (*StartServers*).
- Número de hebras creadas por cada proceso hijo (*ThreadsPerChild*).

- **El MPM *Event***

El MPM *Event* permitirá mostrar y modificar los parámetros de configuración pertenecientes al mismo. Cuenta con las siguientes directivas de configuración:

- Número de procesos hijos que se crean al iniciar Apache (*StartServers*).
- Límite superior del número configurable de procesos (*MaxClients*).
- Número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*).
- Número máximo de hebras en espera (*MaxSpareThreads*).
- Número de hebras creadas por cada proceso hijo (*ThreadsPerChild*).
- Límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*).
- Límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*).

- **Administración y Configuración de directivas Globales**

Este conjunto de directivas permitirá configurar Apache2 de una forma mucho más eficiente, permitiendo mostrar y modificar los siguientes parámetros de configuración cuya función principal es de incrementar el rendimiento del mismo, atender más peticiones simultáneamente y evitar el colapso de los servicios.

- Activar la resolución de DNS de las direcciones IP de los clientes (*HostnameLookups*).
- Buscar ficheros de configuración (*AllowOverrideAll*).

- Permitir que se establezcan conexiones HTTP persistentes (*KeepAlive*).
- Habilitar compresión HTTP.
- Habilitar separador de contenido dinámico y estático

- **Implementación de la Capa de Presentación.**

La interfaz gráfica estará diseñada para brindar al usuario final la posibilidad de iniciar y configurar en su forma más básica el servidor web Apache2 sin necesidad de poseer conocimientos avanzados en la administración y configuración del mismo. Para su implementación se utilizará el patrón de diseño Modelo-Vista-Controlador (MVC), el cual es un patrón o modelo de abstracción de desarrollo de software cuyo principio fundamental es definir una arquitectura con claras responsabilidades para diferenciar sus componentes. Este patrón de diseño permitirá el uso de Spring MVC, el cual es uno de los módulos del *framework* Spring, que provee un exhaustivo soporte para el patrón MVC, facilitando la implementación de la capa de Presentación.

De manera general al implementar las nuevas funcionalidades al módulo Web de HMAST; el mismo contendrá una interfaz gráfica que permitirá configurar y administrar de forma más sencilla las tres secciones de Apache2.

- Sección 1: Entorno global. Se mantienen las funcionalidades de administrar y configurar los puertos, las conexiones, el MPM *prefork*, además se le incluye poder configurar y administrar los MPM *worker*, *event* y *threaded*, así como la configuración de directivas globales las cuales aumentarán el rendimiento del servidor web Apache2.
- Sección 2: Servidor principal. Estas funcionalidades no sufren cambios, se mantiene en esta sección la descripción de la configuración principal del servidor. La configuración seguirá contenida dentro del archivo principal.
- Sección 3: Hosts virtuales. Esta sección de configuración no sufre cambios, se mantiene la gestión de los hosts virtuales, posibilitando adicionar más de un host virtual en el mismo servidor.

### 2.2 Artefactos generados en el desarrollo

En el desarrollo de las nuevas funcionalidades del módulo Web guiado por la metodología SXP, se generaron artefactos como son la Lista de Reserva del Producto (LRP), el Plan de Liberación, las Historias de Usuario y las Tareas de Ingeniería.

### 2.2.1 Requisitos del Sistema

La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software [39]. Ellos se clasifican en:

- Requisitos funcionales: Describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema.
- Requisitos no funcionales: Describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los requisitos funcionales. Los requisitos no funcionales incluyen restricciones cuantitativas, como el tiempo de respuesta o precisión, tipo de plataforma (lenguajes de programación y/o sistemas operativos).

Durante la captura de requisitos se confecciona la LRP, en la cual, se definen las funcionalidades que tendrá el producto en forma de requisitos técnicos y de negocio. Es una lista priorizada y garantiza la organización de los requisitos funcionales y no funcionales, a partir de la prioridad que tengan para el desarrollo del sistema, de igual forma facilita el trabajo en la confección de las Historias de Usuario. A continuación se muestra la tabla tomada del artefacto LRP con los requisitos funcionales y no funcionales determinados para el módulo Web de HMAST.

Prioridad	Ítem *	Descripción	Estimación (semanas)	Estimado por
<b>Alta</b>				
	1.	Instalar Módulo Multiprocesamiento(MPM) <i>Worker</i>	0.2	Nestor A. Delgado Pacheco
	2.	Instalar Módulo Multiprocesamiento(MPM) <i>Event</i>	0.2	
	3.	Instalar Módulo Multiprocesamiento(MPM) <i>Threaded</i>	0.2	
<b>Media</b>				
	4.	Mostrar la directiva <i>StartServers</i> del MPM <i>worker</i> .	0.1	
	5.	Mostrar la directiva <i>MaxClients</i> del MPM <i>worker</i> .	0.1	
	6.	Mostrar la directiva <i>MinSpareThreads</i> del MPM <i>worker</i> .	0.1	
	7.	Mostrar la directiva	0.1	

		<i>MaxSpareThreads</i> del MPM <i>worker</i> .		
8.		Mostrar la directiva <i>ThreadsPerChild</i> del MPM <i>worker</i> .	0.1	
9.		Mostrar la directiva <i>ThreadLimit</i> del MPM <i>worker</i> .	0.1	
10.		Mostrar la directiva <i>MaxRequestsPerChild</i> del MPM <i>worker</i> .	0.1	
11.		Modificar la directiva <i>StartServers</i> del MPM <i>worker</i> .	0.1	
12.		Modificar la directiva <i>MaxClients</i> del MPM <i>worker</i> .	0.1	
13.		Modificar la directiva <i>MinSpareThreads</i> del MPM <i>worker</i> .	0.1	
14.		Modificar la directiva <i>MaxSpareThreads</i> del MPM <i>worker</i> .	0.1	
15.		Modificar la directiva <i>ThreadsPerChild</i> del MPM <i>worker</i> .	0.1	
16.		Modificar la directiva <i>ThreadLimit</i> del MPM <i>worker</i> .	0.1	
17.		Modificar la directiva <i>MaxRequestsPerChild</i> del MPM <i>worker</i> .	0.1	
18.		Mostrar la directiva <i>MaxClients</i> del MPM <i>threaded</i> .	0.1	
19.		Mostrar la directiva <i>MaxRequestsPerChild</i> del MPM <i>threaded</i> .	0.1	
20.		Mostrar la directiva <i>MinSpareThreads</i> del MPM <i>threaded</i> .	0.1	
21.		Mostrar la directiva <i>MaxSpareThreads</i> del MPM <i>threaded</i> .	0.1	
22.		Mostrar la directiva <i>StartServers</i> del MPM <i>threaded</i> .	0.1	
23.		Mostrar la directiva <i>ThreadsPerChild</i> del MPM <i>threaded</i> .	0.1	
24.		Modificar la directiva <i>MaxClients</i> del MPM <i>threaded</i> .	0.1	
25.		Modificar la directiva <i>MaxRequestsPerChild</i> del MPM <i>threaded</i> .	0.1	
26.		Modificar la directiva <i>MinSpareThreads</i> del MPM <i>threaded</i> .	0.1	

27.	Modificar la directiva <i>MaxSpareThreads</i> del MPM <i>threaded</i> .	0.1	
28.	Modificar la directiva <i>StartServers</i> del MPM <i>threaded</i> .	0.1	
29.	Modificar la directiva <i>ThreadsPerChild</i> del MPM <i>threaded</i> .	0.1	
30.	Mostrar la directiva <i>StartServers</i> del MPM <i>event</i> .	0.1	
31.	Mostrar la directiva <i>MaxClients</i> del MPM <i>event</i> .	0.1	
32.	Mostrar la directiva <i>MinSpareThreads</i> del MPM <i>event</i> .	0.1	
33.	Mostrar la directiva <i>MaxSpareThreads</i> del MPM <i>event</i> .	0.1	
34.	Mostrar la directiva <i>ThreadsPerChild</i> del MPM <i>event</i> .	0.1	
35.	Mostrar la directiva <i>ThreadLimit</i> del MPM <i>event</i> .	0.1	
36.	Mostrar la directiva <i>MaxRequestsPerChild</i> del MPM <i>event</i> .	0.1	
37.	Modificar la directiva <i>StartServers</i> del MPM <i>event</i> .	0.1	
38.	Modificar la directiva <i>MaxClients</i> del MPM <i>event</i> .	0.1	
39.	Modificar la directiva <i>MinSpareThreads</i> del MPM <i>event</i> .	0.1	
40.	Modificar la directiva <i>MaxSpareThreads</i> del MPM <i>event</i> .	0.1	
41.	Modificar la directiva <i>ThreadsPerChild</i> del MPM <i>event</i> .	0.1	
42.	Modificar la directiva <i>ThreadLimit</i> del MPM <i>event</i> .	0.1	
43.	Modificar la directiva <i>MaxRequestsPerChild</i> del MPM <i>event</i> .	0.1	
<b>Baja</b>			
44.	Habilitar directiva <i>KeepAlive</i>	0.1	
45.	Deshabilitar directiva <i>KeepAlive</i>	0.1	
46.	Mostrar parámetro de la directiva <i>KeepAliveTimeout</i>	0.1	
47.	Modificar parámetro de la directiva <i>KeepAliveTimeout</i>	0.1	
48.	Habilitar el módulo <i>mod_deflate</i> .	0.1	



49.	Deshabilitar el módulo <i>mod_deflate</i> .	0.1	
50.	Habilitar el módulo <i>mod_proxy</i>	0.1	
51.	Deshabilitar el módulo <i>mod_proxy</i>	0.1	
52.	Habilitar el módulo <i>rewrite_module</i>	0.1	
53.	Deshabilitar el módulo <i>rewrite_module</i>	0.1	
54.	Mostrar parámetro de la directiva <i>TimeOut</i>	0.1	
55.	Modificar parámetro de la directiva <i>TimeOut</i>	0.1	
<b>RNF (Requisitos No Funcionales)</b>			
<b>Software</b>			
1.	La aplicación se ejecutará en los sistemas operativos Nova, Ubuntu o Debian.		
2.	Usar como lenguaje de programación Java.		
3.	El IDE a utilizar será NetBeans 7.3.		
4.	Usar como framework de desarrollo Spring.		
5.	Usar Augeas para la configuración de ficheros.		
<b>Diseño</b>			
6.	Crear interfaz sencilla e intuitiva desde la cual se acceda a todas las funcionalidades del módulo.		

### 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. A continuación se muestran las Historias de Usuario de prioridad alta y media, la de prioridad baja se puede consultar en el Anexo 5.

<b>Historia de Usuario</b>	
<b>Número:</b>	<b>Nombre Historia de Usuario:</b> Instalar Módulos Multiprocesamiento de Apache2.

HMAST_WEB_1	
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Nestor A. Delgado Pacheco	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1 semana
<b>Riesgo en Desarrollo:</b> Alta	<b>Puntos Reales:</b> 1 semana
<b>Descripción:</b> Permitirá instalar el Módulo de Multiprocesamiento deseado: <i>worker, event y threaded</i> .	
<b>Observaciones:</b> Cuando se accede al módulo Web en la pestaña de Módulos de Multiprocesamiento (MPM) se podrá instalar el MPM deseado y posteriormente administrarlo. Solo podrá estar instalado un MPM en el módulo Web en un determinado momento.	
<b>Prototipo de interface:</b> Ver Anexo 1: Instalar Módulos Multiprocesamiento de Apache2.	

Historia de Usuario	
<b>Número:</b> HMAST_WEB_2	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Nestor A. Delgado Pacheco	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Media	<b>Puntos Reales:</b> 2 semanas
<b>Descripción:</b> Permitirá mostrar y modificar las parámetros de configuración pertenecientes al MPM <i>worker</i> . Este Módulo de Multiprocesamiento (MPM) implementa un servidor híbrido multiproceso-multihebra. Usando hilos para atender peticiones, el servidor puede servir un mayor número de peticiones con menos recursos de sistema que un servidor basado únicamente en procesos. No obstante, se mantiene casi por completo la estabilidad de un servidor basado en procesos manteniendo la capacidad multiproceso, pudiendo cada proceso tener muchas hebras.	
<b>Observaciones:</b> El software permitirá al usuario mostrar y modificar los parámetros que se encuentran en el fichero de administración principal <i>apache2.conf</i> ubicado en <i>etc/apache2</i> . Los valores que	

pueden tomar las distintas directivas deben ser números enteros positivos.

- **Modificar número de procesos hijos que se crean al iniciar Apache (*StartServers*):** Esta directiva especifica el número de procesos hijos que se crean al iniciar Apache2. El valor cambia según el MPM que se trate, para *worker* el valor por defecto es 2 y máximo es 3.
- **Mostrar número de procesos hijos que se crean al iniciar Apache (*StartServer*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite superior del número configurable de procesos (*MaxClients*):** Especifica el límite de peticiones simultáneas que serán atendidas. Cualquier intento de conexión por encima del límite *MaxClients* se pondrá en cola. Para este tipo de MPM, *MaxClients* limita el número total de hebras que van a estar disponibles para servir clientes. Su valor máximo y por defecto es 150.
- **Mostrar límite superior del número configurable de procesos (*MaxClients*):** Permite mostrar el valor actual de la directiva.
- **Modificar número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*):** Número mínimo de hebras en espera para atender picos de demanda en las peticiones. Los diferentes MPM tratan esta directiva de forma diferente. El valor por defecto es 25 y máximo para este MPM es de 75.
- **Mostrar número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número máximo de hebras en espera (*MaxSpareThreads*):** Número máximo de hebras en espera. Los diferentes MPM tienen diferentes comportamientos respecto a esta directiva. El valor por defecto es 75 y máximo para este MPM es de 250. Este MPM monitoriza el número de hebras en espera en el servidor en conjunto. Si hay demasiadas hebras en espera en el servidor, se eliminan algunos procesos hijos hasta que el número de hebras en espera se ajuste al límite especificado.
- **Mostrar número máximo de hebras en espera (*MaxSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número de hebras creadas por cada proceso hijo (*ThreadsPerChild*):** Esta directiva especifica el número de hebras creadas por cada proceso hijo. El proceso hijo crea estas hebras al inicio y no vuelve a crear más. En este MPM que hay múltiples procesos hijos, el número total de hebras debería ser lo suficientemente grande como para atender la carga en circunstancias normales del servidor. El valor por defecto y máximo de esta directiva para este

MPM es de 25.

- **Mostrar número de hebras creadas por cada proceso hijo (*ThreadsPerChild*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*):** Esta directiva determina el valor máximo que puede especificarse en la directiva *ThreadsPerChild* para el tiempo de vida de un proceso de Apache2. Los intentos por modificar este valor durante un reinicio serán ignorados, pero el valor de la directiva *ThreadsPerChild* puede modificarse durante un reinicio hasta un valor igual al de esta directiva. Si en la directiva *ThreadLimit* se especifica un valor mucho más grande que en *ThreadsPerChild*, se reservará memoria compartida en exceso que no será usada. Si tanto en *ThreadLimit* como en *ThreadsPerChild* se especifican valores mayores de los que el sistema puede tratar, Apache podría no iniciarse o su funcionamiento podría volverse inestable. El valor por defecto y máximo de esta directiva para este MPM es 64.
- **Mostrar límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*):** La directiva *MaxRequestsPerChild* especifica el número máximo de peticiones que un proceso hijo atenderá durante su existencia. Después de atender *MaxRequestsPerChild* peticiones, el proceso hijo se eliminará. Si el valor especificado en esta directiva *MaxRequestsPerChild* es 0, no habrá límite. El valor por defecto es 0 y máximo de esta directiva es 1000.
- **Mostrar límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*):** Permite mostrar el valor actual de la directiva.

**Prototipo de interface:** Ver Anexo 2: Gestionar parámetros en el MPM *worker* de Apache2.

#### Historia de Usuario

<b>Número:</b> HMAST_WEB_3	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>threaded</i> de Apache2
-------------------------------	--

**Modificación de Historia de Usuario Número: 1**

<b>Usuario:</b> Nestor A. Delgado Pacheco	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Media	<b>Puntos Reales:</b> 2 semanas
<p><b>Descripción.</b> Permitirá mostrar y modificar los parámetros de configuración pertenecientes al MPM <i>threaded</i>. Este MPM permite el soporte de hilos en Apache2. Es como el MPM <i>prefork</i>, pero en vez de que cada proceso hijo tenga un solo hilo, cada proceso hijo puede tener un número determinado de hilos. Si Apache2 inicia 30 procesos hijos en los que cada hijo puede tener un máximo de 10 hilos, entonces Apache2 podrá servir <math>30 \times 10 = 300</math> solicitudes simultáneas. Dado que los hilos son más eficaces en cuanto a recursos que los procesos este MPM es muy escalable.</p>	
<p><b>Observaciones:</b> El software permitirá al usuario mostrar y modificar los parámetros que se muestran a continuación. Los valores que pueden tomar las distintas directivas deben ser números enteros positivos.</p> <ul style="list-style-type: none"> <li>• <b>Modificar límite superior del número configurable de procesos (<i>MaxClients</i>):</b> Especifica el límite de peticiones simultáneas que serán atendidas. Como Apache2 utiliza un servidor hijo para cada solicitud, este es también el límite efectivo para el número de servidores hijos que pueden existir al mismo tiempo. El valor por defecto y máximo para este MPM es de 256.</li> <li>• <b>Mostrar límite superior del número configurable de procesos (<i>MaxClients</i>):</b> Permite mostrar el valor actual de la directiva.</li> <li>• <b>Modificar límite en el número de peticiones que un proceso hijo puede atender durante su vida (<i>MaxRequestsPerChild</i>):</b> La directiva <i>MaxRequestsPerChild</i> especifica el número máximo de peticiones que un proceso hijo atenderá durante su existencia. Después de atender <i>MaxRequestsPerChild</i> peticiones, el proceso hijo se eliminará. Si el valor especificado en esta directiva <i>MaxRequestsPerChild</i> es 0, no habrá límite. El valor para este MPM por defecto es 0.</li> <li>• <b>Mostrar límite en el número de peticiones que un proceso hijo puede atender durante su vida (<i>MaxRequestsPerChild</i>):</b> Permite mostrar el valor actual de la directiva.</li> <li>• <b>Modificar número máximo de hilos en espera (<i>MaxSpareThreads</i>):</b> La directiva <i>MaxSpareThreads</i> fija el número máximo de hilos en espera. El MPM <i>threaded</i> trata con los hilos en espera en la base de un servidor, que significa que si hay demasiados hilos en espera en el servidor, comienza a destruir procesos hijos hasta que el número de hilos en espera es menor que el número que se especifica en este parámetro. El valor por defecto y máximo para</li> </ul>	

este MPM es de 500.

- **Mostrar número máximo de hilos en espera (*MaxSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número mínimo de hilos en espera (*MinSpareThreads*):** Determina el número de hilos en espera. Este MPM negocia con hilos en espera en la base de un servidor, lo que significa que cuando hay menos hilos en espera que el número especificado en este parámetro, Apache2 crea nuevos procesos hijos para alcanzar este número. El valor por defecto y máximo para este MPM es de 250.
- **Mostrar número mínimo de hilos en espera (*MinSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número de procesos hijos que se crean al iniciar Apache (*StartServers*):** Esta directiva especifica el número de procesos hijos que se crean al iniciar Apache2. El valor por defecto y máximo cambia según el MPM de que se trate, para este MPM es de 5.
- **Mostrar número de procesos hijos que se crean al iniciar Apache (*StartServers*):** Permite mostrar el valor actual de la directiva.
- **Modificar número de hilos creados por cada proceso hijo (*ThreadsPerChild*):** Esta directiva le dice al servidor cuántos hilos debería utilizar para cada proceso hijo. Su valor por defecto y máximo para este MPM es de 50.
- **Mostrar número de hilos creados por cada proceso hijo (*ThreadsPerChild*):** Permite mostrar el valor actual de la directiva.

**Prototipo de interface:** Ver Anexo 3: Gestionar parámetros en el MPM *threaded* de Apache2.

Historia de Usuario	
<b>Número:</b> HMAST_WEB_4	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
<b>Modificación de Historia de Usuario Número: 1</b>	
<b>Usuario:</b> Nestor A. Delgado Pacheco	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2 semanas
<b>Riesgo en Desarrollo:</b> Media	<b>Puntos Reales:</b> 2 semanas

**Descripción:** Permitirá mostrar y modificar los parámetros de configuración pertenecientes al MPM *event*. Se basa en el MPM *worker* y sus directivas son iguales. Intenta solucionar el problema HTTP de *KeepAlive*. Cuando un cliente completa la primera petición, el cliente puede mantener abierta la conexión y enviar más peticiones usando el mismo *socket*. Esto puede mejorar el rendimiento en la creación de conexiones TCP, sin embargo, Apache2 tradicionalmente mantiene un hijo completo *process/thread* esperando los datos de un cliente lo que tiene sus propias desventajas. Para resolver este problema MPM *event* usa un único *thread* dedicado a manejar tanto los *sockets* de escucha como todos los *sockets* que están en un estado *KeepAlive*.

**Observaciones:** El software permitirá al usuario mostrar y modificar los siguientes parámetros los cuales están ubicados en el fichero de administración principal *apache2.conf* ubicado en *etc/apache2*. Los valores que pueden tomar las distintas directivas deben ser números enteros positivos.

- **Modificar número de procesos hijos que se crean al iniciar Apache (*StartServers*):** Esta directiva especifica el número de procesos hijos que se crean al iniciar Apache2. El valor cambia según el MPM de que se trate, para *event* el valor por defecto y máximo es 2.
- **Mostrar número de procesos hijos que se crean al iniciar Apache (*StartServer*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite superior del número configurable de procesos (*MaxClients*):** Especifica el límite de peticiones simultáneas que serán atendidas. Cualquier intento de conexión por encima del límite *MaxClients* se pondrá en cola. Para este tipo de MPM, *MaxClients* limita el número total de hebras que van a estar disponibles para servir clientes. Su valor máximo y por defecto es 150.
- **Mostrar límite superior del número configurable de procesos (*MaxClients*):** Permite mostrar el valor actual de la directiva.
- **Modificar número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*):** Los diferentes MPM tratan esta directiva de forma diferente. El valor por defecto y máximo para este MPM es 25.
- **Mostrar número mínimo de hebras en espera para atender picos de demanda en las peticiones (*MinSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número máximo de hebras en espera (*MaxSpareThreads*):** Los diferentes MPM tienen diferentes comportamientos respecto a esta directiva. El valor por defecto y máximo para este MPM es de 75. Este MPM monitoriza el número de hebras en espera en el servidor en conjunto. Si hay demasiadas hebras en espera en el servidor, se eliminan algunos procesos

hijos hasta que el número de hebras en espera se ajuste al límite especificado.

- **Mostrar número máximo de hebras en espera (*MaxSpareThreads*):** Permite mostrar el valor actual de la directiva.
- **Modificar número de hebras creadas por cada proceso hijo (*ThreadsPerChild*):** Esta directiva especifica el número de hebras creadas por cada proceso hijo. El proceso hijo crea estas hebras al inicio y no vuelve a crear más. En este MPM que hay múltiples procesos hijos, el número total de hebras debería ser lo suficientemente grande como para atender la carga en circunstancias normales del servidor. El valor por defecto y máximo de esta directiva para este MPM es de 25.
- **Mostrar número de hebras creadas por cada proceso hijo (*ThreadsPerChild*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*):** Esta directiva determina el valor máximo que puede especificarse en la directiva *ThreadsPerChild* para el tiempo de vida de un proceso de Apache2. Los intentos por modificar este valor durante un reinicio serán ignorados, pero el valor de la directiva *ThreadsPerChild* puede modificarse durante un reinicio hasta un valor igual al de la directiva *ThreadLimit*. Si en la directiva *ThreadLimit* se especifica un valor mucho más grande que en *ThreadsPerChild*, se reservará memoria compartida en exceso que no será usada. Si tanto en *ThreadLimit* como en *ThreadsPerChild* se especifican valores mayores de los que el sistema puede tratar, Apache2 podría no iniciarse o su funcionamiento podría volverse inestable. El valor por defecto y máximo de esta directiva para este MPM es 64.
- **Mostrar límite superior del número de hebras por proceso hijo que pueden especificarse (*ThreadLimit*):** Permite mostrar el valor actual de la directiva.
- **Modificar límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*):** La directiva *MaxRequestsPerChild* especifica el número máximo de peticiones que un proceso hijo atenderá durante su existencia. Después de atender *MaxRequestsPerChild* peticiones, el proceso hijo se eliminará. Si el valor especificado en esta directiva *MaxRequestsPerChild* es 0, no habrá límite. El valor por defecto es 0 y máximo de esta directiva es 1000.
- **Mostrar límite en el número de peticiones que un proceso hijo puede atender durante su vida (*MaxRequestsPerChild*):** Permite mostrar el valor actual de la directiva.

**Prototipo de interface:** Ver Anexo 4: Gestionar parámetros en el MPM *event* de Apache2.



### 2.3 Arquitectura del Módulo

En el desarrollo de las nuevas funcionalidades del módulo Web se hace uso de la arquitectura N-Capas explicada en el Capítulo 1 Epígrafe 1.4.1, la cual está en correspondencia con la arquitectura propuesta por la herramienta HMAST. El alcance del desarrollo de nuevas funcionalidades incluye la interfaz visual, por lo que se implementará la capa de Presentación. Se mantienen las capas de Aplicación, Dominio y Persistencia con un paquete con todo lo referente al módulo y la capa Infraestructura Transversal con todas las clases reutilizables, como se aprecia en la Figura 2.1.

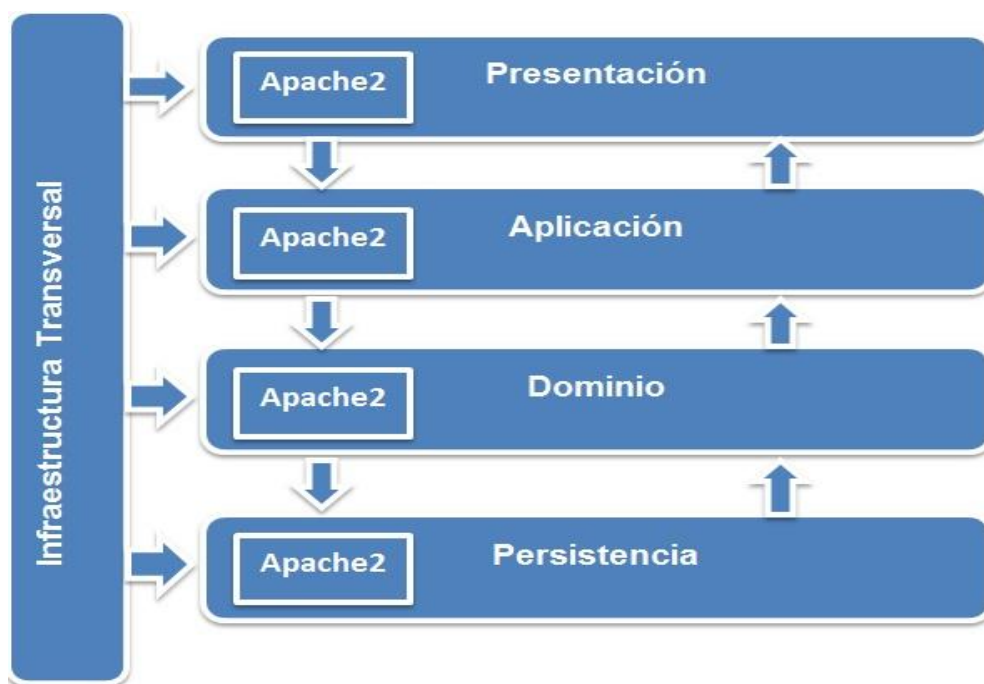


Figura 2.1: Arquitectura del Módulo Web incluyendo la capa de Presentación.

### 2.4 Diagrama de paquetes

Para la representación de las nuevas funcionalidades del módulo Web basado en la organización de paquetes y sus elementos, se diseñó el diagrama de paquetes tomando como marco de referencia la arquitectura propuesta. Existe un paquete representando cada capa y dentro de este uno llamado Apache2 que contiene todo lo referente al módulo.

Las nuevas funcionalidades del módulo Web inciden en la sección *GlobalConfiguration* que contiene todo lo referente a la gestión de los MPM.

En las siguientes figuras 2.2, 2.3, 2.4, 2.5 y 2.6 se representan las diferentes partes del diagrama de

paquetes perteneciente a la historia de usuario **Gestionar parámetros del MPM worker de Apache2**, perteneciente a *GlobalConfiguration*. A cada grupo corresponde un flujo de acciones similares independientemente de su funcionalidad.

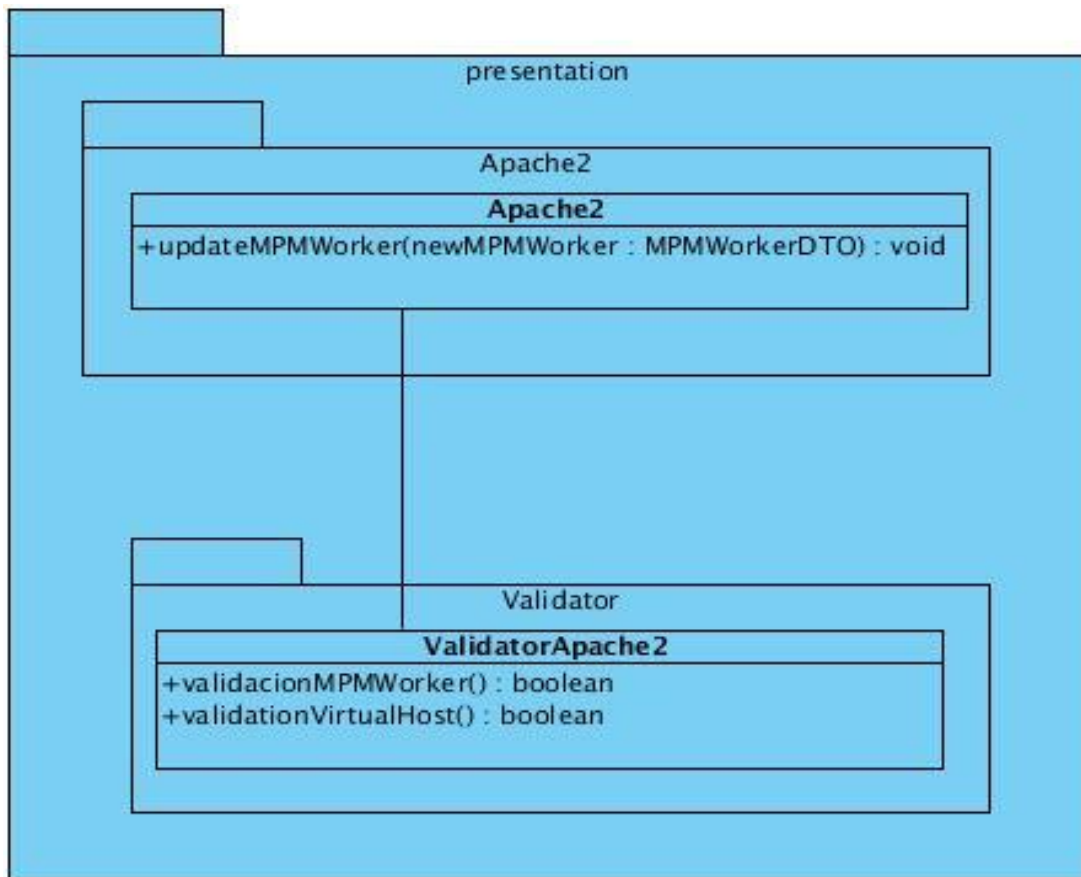


Figura 2.2: Diagrama de paquetes correspondiente a la capa de Presentación.

En la Figura 2.2 se representa la capa de Presentación, en el paquete Apache2 se encuentra la clase *Apache2* la cual captura la información entrada por el usuario y hace las peticiones a la capa inferior mostrando al usuario la respuesta proveniente de esta. Además se encuentra el paquete *Validator* el cual contiene la clase *ValidatorApache2* que contiene las validaciones a nivel de interface. *Apache2* posee una relación de asociación con *ValidatorApache2* que se realiza mediante una inyección de dependencia.

Las capas de Presentación y Aplicación se relacionan debido a que *Apache2* posee una relación de asociación con *IGlobalConfigurationAppService* y con *IVirtualHostAppService* que se realiza mediante una inyección de dependencia.

En la Figura 2.3 se representa la capa de Aplicación, en el paquete Apache2 existe una interface *IGlobalConfigurationAppService* que define los métodos que serán llamados desde la capa de Presentación y una clase *GlobalConfigurationAppService* que los implementa. En Apache2 se encuentra también un paquete *DTO* que contiene los objetos de transferencia de datos que serán enviados desde y hacia la capa de presentación.

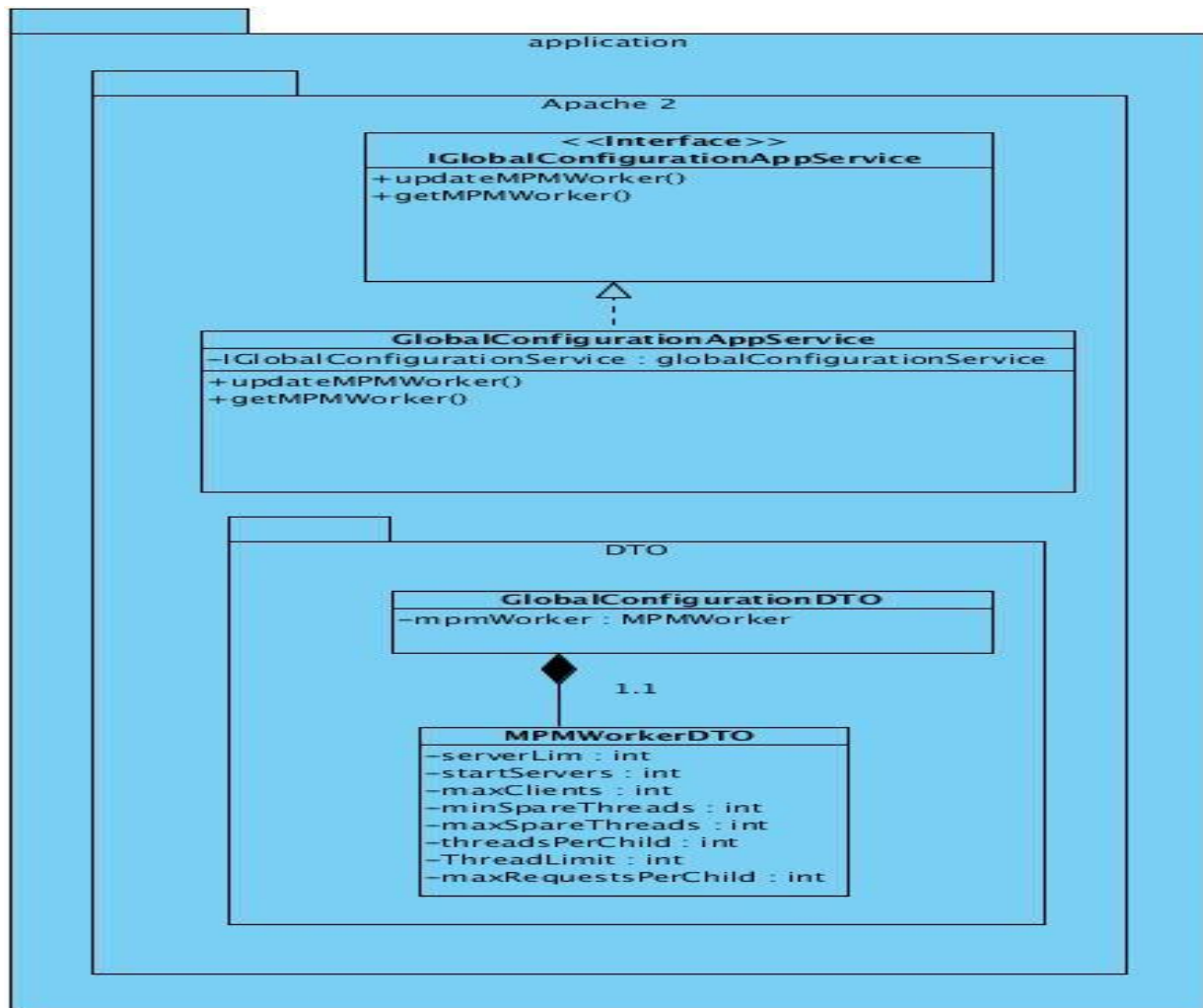


Figura 2.3: Diagrama de paquetes correspondiente a la capa de Aplicación.

Las capas de Aplicación y Dominio se relacionan debido a que *GlobalConfigurationAppService* posee una relación de asociación con *IGlobalConfigurationService*, que se realiza mediante una inyección de dependencia.

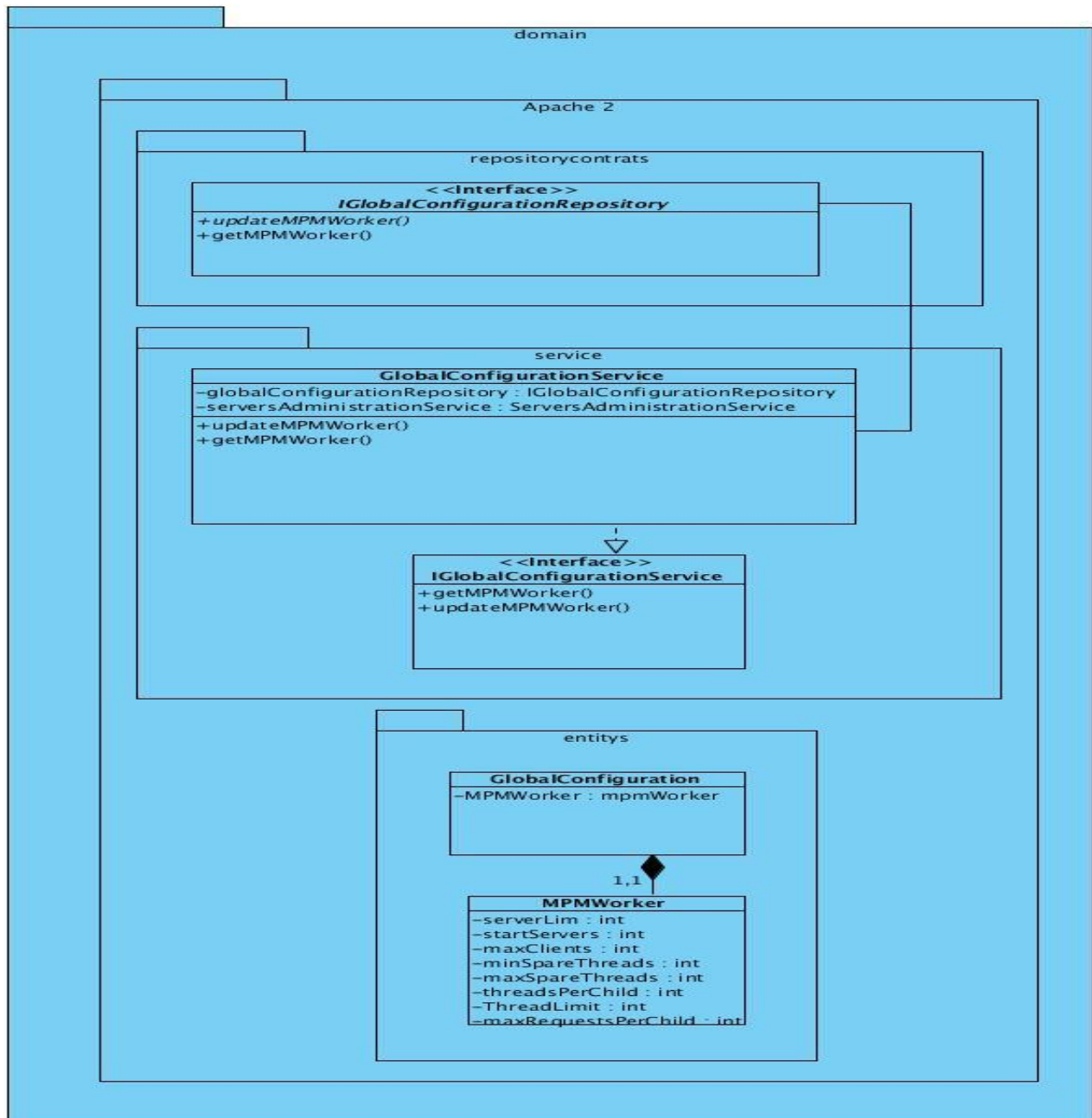


Figura 2.4: Diagrama de paquetes correspondiente a la capa de Dominio.

La capa de Dominio está representada en la Figura 2.4 y contiene dentro del paquete Apache2 tres paquetes llamados *entitys*, *services* y *repositorycontrats*.

- *entitys*: contiene las entidades *MPMWorker* y *GlobalConfiguration*. En cada una de ellas se realizan validaciones atómicas para cada atributo.

- *services*: se realizan las validaciones que no se realizaron en las entidades. Contiene la interface *IGlobalConfigurationService* que define las funcionalidades y *GlobalConfigurationService* que las implementa. *GlobalConfigurationService* posee una relación de asociación con *IGlobalConfigurationRepository* que se realiza mediante una inyección de dependencia.
- *repositorycontrats*: define los contratos de repositorio en *IGlobalConfigurationRepository* pero no los implementa.

El paquete *Apache2* correspondiente a la capa de Persistencia, como se aprecia en la Figura 2.5, contiene otro paquete llamado *repository*, en el cual se encuentra la clase *GlobalConfigurationRepository* que es la encargada de realizar la persistencia de los datos implementando los contratos de repositorio definidos en *IGlobalConfigurationRepository*. La clase *GlobalConfigurationRepository* posee una relación de dependencia con la entidad *GlobalConfiguration* debido a que contiene un mapa que almacena objetos que son instancias de la misma.

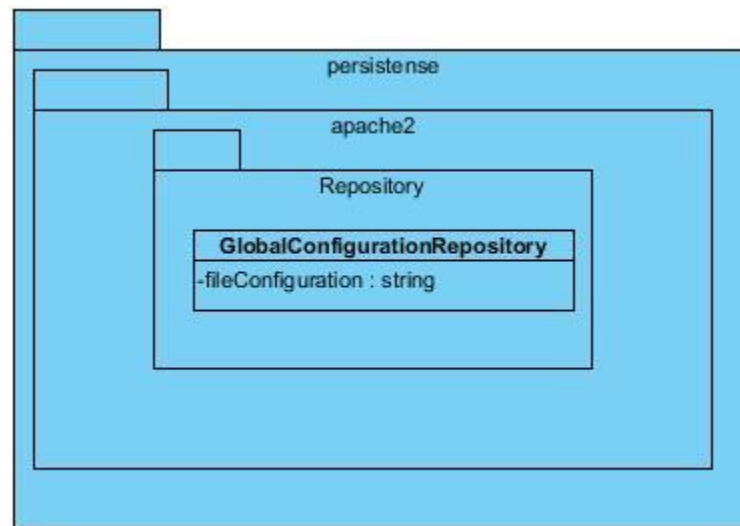


Figura 2.5: Diagrama de paquetes correspondiente a la capa de Persistencia.

La Figura 2.6 representa el diagrama de paquetes correspondiente a la capa de Infraestructura Transversal, que contiene la entidad *Entity* la cual es padre de todas las entidades definidas en la capa de Dominio. En esta capa, en caso necesario se incluyen además clases que sean reutilizables por otras capas.

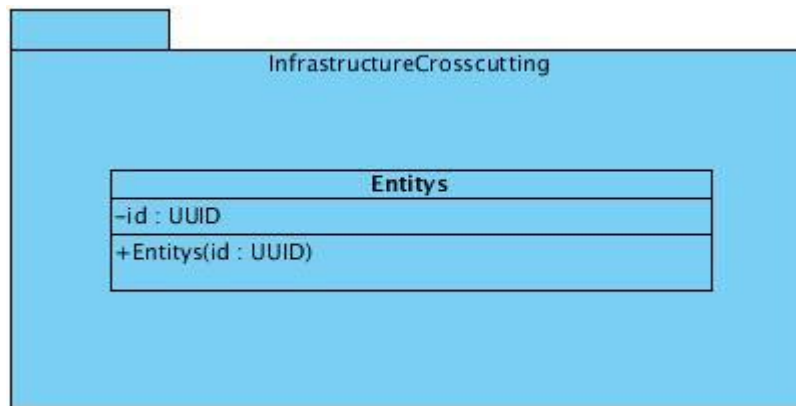


Figura 2.6: Diagrama de paquetes correspondiente a la capa de Infraestructura Transversal.

## 2.5 Patrones de diseño empleados

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. En el desarrollo de nuevas funcionalidades al módulo Web de HMAST se utilizan los patrones GRASP y GOF.

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresado en forma de patrones. Para el desarrollo del módulo se hace uso de los siguientes patrones GRASP.

**Creador:** Se pone de manifiesto en la capa de Aplicación cuando se realiza 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. Igualmente se manifiesta en la conversión de entidades a DTO.

**Experto:** La principal aplicación de este patrón es que cada capa tiene un objetivo y una responsabilidad asignada, siendo solo ella la encargada de implementarla.

**Bajo Acoplamiento:** Se pone de manifiesto en las inyecciones de dependencia proporcionadas por el *framework* Spring lo que permiten el uso de este patrón en el sistema. Los objetos no crean o buscan sus dependencias sino que estas son dadas al objeto.

**Alta Cohesión:** La cohesión es una medida de cuán enfocadas están las responsabilidades de las clases. Se manifiesta en el módulo en la parte de las validaciones, las mismas se realizan en las entidades a sus atributos primitivos y las restantes son realizadas en la capa de Dominio en el paquete de servicio.

Los patrones GOF (*The Gang of Four*) describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos, los mismos solucionan problemas de creación de instancias, ayudando a encapsular y abstraer dicha creación. El patrón GOF utilizado es el:

**Singleton:** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos. Se pone de manifiesto en las clases *SSHConexcion* y *AugeasSingleton* ya que al realizar una conexión al servidor se crea una instancia de las mismas.

### **Conclusiones Parciales**

En el presente capítulo a partir del estudio realizado en el Capítulo 1 y del funcionamiento de Apache2, se definieron un total de 60 funcionalidades a implementar, las cuales están organizadas en 5 Historias de Usuario. Se realizó la descripción de las Historias de Usuario lo que permitió un mejor entendimiento entre el programador y el cliente. Se explicó la arquitectura del módulo Web tomándose esta como base para el diseño del diagrama de paquetes, lo que permitió describir los patrones de diseño utilizados así como conocer la relación entre las capas.

### Capítulo 3: “Implementación y pruebas de las nuevas funcionalidades del módulo Web para HMAST”

Los procesos de desarrollo de software implican la realización de una serie de actividades que garanticen la calidad del mismo. Es por ello que primeramente se realiza la implementación y posterior a esta se define una etapa de pruebas con el objetivo de mostrar errores no detectados hasta entonces. En el presente capítulo se definen: la planificación de la implementación, las Tareas de Ingeniería como apoyo al proceso de implementación y se describen las pruebas realizadas para la verificación de la calidad del módulo implementado.

#### 3.1 Planificación de la Implementación

El Plan de liberación es el artefacto que se obtiene al planificar la implementación, contiene las iteraciones que se van a realizar y las características correspondientes a cada una de ellas. Se definen las Historias de Usuario que se van a desarrollar en cada iteración y la estimación del tiempo en semanas. A continuación se muestra la tabla correspondiente al Plan de liberación.

Liberación	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Dar cumplimiento a las Historias de Usuario de prioridad Alta.	1	1 Semana
2	Dar cumplimiento a las Historias de Usuario de prioridad Media.	2,3,4	6 Semanas
3	Dar cumplimiento a las Historias de Usuario de prioridad Baja.	5	1 Semana

#### 3.2 Tareas de Ingeniería

Las Tareas de Ingeniería se generan como artefacto de la metodología SXP, facilitando el entendimiento en el proceso de implementación debido a que organizan el mismo con la definición de actividades asociadas a las Historias de Usuario. A continuación se definen las Tareas de Ingeniería correspondientes a las Historias de Usuario de prioridad alta y media especificadas en el capítulo anterior, las restantes se pueden consultar en el Anexo 6.



- **Tarea de Ingeniería para la Historia de Usuario Instalar Módulos de Multiprocesamiento de Apache2.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HMAST_WEB_1
<b>Nombre Tarea:</b> Probar manejo de estados en los MPM de Apache2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco	
<p><b>Descripción:</b> Se realizan acciones para el manejo de estados de los MPM de Apache2. Estas acciones no están directamente relacionadas con el fichero de configuración, sino que se realizan en la consola con los siguientes comandos:</p> <ul style="list-style-type: none"> <li>• Instalar MPM <i>Prefork</i>: apt-get install apache2-mpm-prefork</li> <li>• Instalar MPM <i>Worker</i>: apt-get install apache2-mpm-worker</li> <li>• Instalar MPM <i>Event</i>: apt-get install apache2-mpm-event</li> <li>• Instalar MPM <i>Threaded</i>: apt-get install apache2-mpm-threaded</li> </ul>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> HMAST_WEB_1
<b>Nombre Tarea:</b> Diseño de interfaz para manejo de estados en los MPM de Apache2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco	
<p><b>Descripción:</b> Se diseña una interfaz que brinde la posibilidad de instalar los MPM <i>prefork</i>, <i>worker</i> y <i>event</i> de Apache2, además de posibilitar la opción de administrar el MPM instalado. Se muestra una imagen animada la cual indica que se está realizando el proceso de instalación.</p>	

- **Tarea de Ingeniería para la Historia de Usuario Gestionar parámetros en el MPM *worker* de Apache2.**

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> HMAST_WEB_2
<b>Nombre Tarea:</b> Probar funcionamiento de directivas en el MPM <i>worker</i> de Apache2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco	
<p><b>Descripción:</b> Se verifica el funcionamiento de directivas en el MPM <i>worker</i> de Apache2. Su configuración se realiza en el fichero de configuración principal <code>/etc/apache2/apache2.conf</code> específicamente dentro de la directiva de bloque <code>&lt;IfModule mpm_worker_module&gt;&lt;/IfModule&gt;</code>. Las directivas son <i>StartServers</i>, <i>MaxClients</i>, <i>MinSpareThreads</i>, <i>MaxSpareThreads</i>, <i>ThreadsPerChild</i>, <i>ThreadLimit</i>, <i>MaxRequestsPerChild</i>.</p>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 4	<b>Número Historia de Usuario:</b> HMAST_WEB_2
<b>Nombre Tarea:</b> Diseño de interfaz para administrar y configurar las directivas en el MPM <i>worker</i> de Apache2.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco	
<p><b>Descripción:</b> Se diseña una interfaz que brinde la posibilidad de administrar y configurar las directivas <i>StartServers</i>, <i>MaxClients</i>, <i>MinSpareThreads</i>, <i>MaxSpareThreads</i>, <i>ThreadsPerChild</i>, <i>ThreadLimit</i>, <i>MaxRequestsPerChild</i> del MPM <i>worker</i> de Apache2. Además cuenta con un botón para modificar los valores de las directivas y otro para guardar los cambios realizados en los mismos.</p>	

- **Tarea de Ingeniería para la Historia de Usuario Gestionar parámetros en el MPM *threaded* de Apache2.**

Tarea de Ingeniería
---------------------

<b>Número Tarea:</b> 5		<b>Número Historia de Usuario:</b> HMAST_WEB_3	
<b>Nombre Tarea:</b> Probar funcionamiento de directivas en el MPM <i>threaded</i> de Apache2.			
<b>Tipo de Tarea:</b> Desarrollo		<b>Puntos Estimados:</b> 0.2	
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco			
<p><b>Descripción:</b> Se verifica el funcionamiento de directivas en el MPM <i>threaded</i> de Apache2. Su configuración se realiza en el fichero de configuración principal <code>/etc/apache2/apache2.conf</code> específicamente dentro de la directiva de bloque <code>&lt;IfModule mpm_threaded_module&gt;&lt;/IfModule&gt;</code>. Las directivas son <i>MaxClients</i>, <i>MaxRequestsPerChild</i>, <i>MaxSpareThreads</i>, <i>MinSpareThreads</i>, <i>StartServers</i>, <i>ThreadsPerChild</i>.</p>			

Tarea de Ingeniería			
<b>Número Tarea:</b> 6		<b>Número Historia de Usuario:</b> HMAST_WEB_3	
<b>Nombre Tarea:</b> Diseño de interfaz para administrar y configurar las directivas en el MPM <i>threaded</i> de Apache2.			
<b>Tipo de Tarea:</b> Desarrollo		<b>Puntos Estimados:</b> 0.2	
<b>Programador Responsable:</b> Nestor A. Delgado Pacheco			
<p><b>Descripción:</b> Se diseña una interfaz que brinde la posibilidad de administrar y configurar las directivas <i>MaxClients</i>, <i>MaxRequestsPerChild</i>, <i>MaxSpareThreads</i>, <i>MinSpareThreads</i>, <i>StartServers</i>, <i>ThreadsPerChild</i> del MPM <i>threaded</i> de Apache2. Además cuenta con un botón para modificar los valores de las directivas y otro para guardar los cambios realizados en los mismos.</p>			

- **Tarea de Ingeniería para la Historia de Usuario Gestionar parámetros en el MPM *event* de Apache2.**

Tarea de Ingeniería			
<b>Número Tarea:</b> 7		<b>Número Historia de Usuario:</b> HMAST_WEB_4	
<b>Nombre Tarea:</b> Probar funcionamiento de directivas en el MPM <i>event</i> de Apache2.			
<b>Tipo de Tarea:</b> Desarrollo		<b>Puntos Estimados:</b> 0.2	

**Programador Responsable:** Nestor A. Delgado Pacheco

**Descripción:** Se verifica el funcionamiento de directivas en el MPM *event* de Apache2. Su configuración se realiza en el fichero de configuración principal */etc/apache2/apache2.conf* específicamente dentro de la directiva de bloque `<IfModule mpm_event_module></IfModule>`. Las directivas son *StartServers*, *MaxClients*, *MinSpareThreads*, *MaxSpareThreads*, *ThreadsPerChild*, *ThreadLimit*, *MaxRequestsPerChild*.

#### Tarea de Ingeniería

**Número Tarea:** 8

**Número Historia de Usuario:** HMAST\_WEB\_4

**Nombre Tarea:** Diseño de interfaz para administrar y configurar las directivas en el MPM *event* de Apache2.

**Tipo de Tarea:** Desarrollo

**Puntos Estimados:** 0.2

**Programador Responsable:** Nestor A. Delgado Pacheco

**Descripción:** Se diseña una interfaz que brinde la posibilidad de administrar y configurar las directivas *StartServers*, *MaxClients*, *MinSpareThreads*, *MaxSpareThreads*, *ThreadsPerChild*, *ThreadLimit*, *MaxRequestsPerChild* del MPM *event* de Apache2. Además cuenta con un botón para modificar los valores de las directivas y otro para guardar los cambios realizados en los mismos.

### 3.3 Estándares de codificación

El propósito fundamental de los estándares de codificación es tener una arquitectura y un estilo consistente, independiente del autor, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. El mayor problema que trata de enfrentar esta práctica, es el tratar de entender el formato y el estilo utilizado en el código escrito por otros desarrolladores [40].

#### Estándar para nombrar las clases

Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto se pondrá con minúscula, cuando sea un nombre compuesto se utilizará la notación Pascal.

Ejemplo: *GlobalConfiguration*

### **Estándar para nombrar las funciones**

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se utiliza la notación *Camel* y con sólo leerlo se reconoce el propósito de la misma.

Ejemplo: *saveFileConfiguration*

### **Estándar para nombrar las variables**

En el nombre de las variables se escribe la primera palabra con minúscula, si es un nombre compuesto se utilizará notación *Camel*.

Ejemplo: *virtualHost*

### **Estándar para nombrar los componentes**

Todos los paquetes comienzan con `cu.uci.hmast.xxx.yyy.zzz.kkk`

`xxx` → `presentation`, `application`, `domain`, `persistence`.

`yyy` → nombre del módulo (`squid3`, `dhcp3`, `apache2`).

`zzz` → elementos que pueden contener los componentes verticales (`entitys`, `repositorys`).

`kkk` → subpaquetes.

Ejemplo: `cu.uci.hmast.domain.apache2.entitys`

## **3.4 Pruebas al Módulo Web**

La realización de pruebas es la tarea de demostrar que un programa realiza las funciones para las cuales fue construido. Las pruebas no aseguran la ausencia de errores, sino que están orientadas a demostrar que existen defectos en el software. Las mismas se dividen según el método, en pruebas de “Caja blanca” que se centran en los detalles procedimentales y en “Caja negra” que se aplican a la interfaz. El producto desarrollado cuenta con una interfaz gráfica por lo que se le aplicaron pruebas de Aceptación (caja negra) que es el tipo de prueba definido en la metodología SXP.

### **3.4.1 Pruebas de Aceptación**

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponde al cliente. La utilización de estas permite a los

programadores medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente. Debido al extenso número de pruebas realizadas, solo se exponen 4 de los 59 casos de prueba de aceptación que tiene el presente trabajo de diploma. Las restantes pruebas de aceptación pueden consultarse en el Anexo 7.

Caso de Prueba de Aceptación			
<b>Código</b>	<b>Caso de Prueba:</b>	<b>Nombre Historia de Usuario:</b>	Instalar Módulos
<i>HMAST_Web_1-</i>		Multiprocesamiento de Apache2.	
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco			
<b>Descripción de la Prueba:</b> Probar que se instala el MPM <i>prefork</i> de Apache2.			
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.			
<b>Entrada / Pasos de ejecución:</b>			
<ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Seleccionar el MPM <i>prefork</i>.</li> </ul>			
<b>Resultado Esperado:</b>			
El MPM <i>prefork</i> se instala y se muestra un mensaje indicando su correcta instalación.			
<b>Evaluación de la Prueba:</b> Satisfactoria			

Caso de Prueba de Aceptación			
<b>Código</b>	<b>Caso de Prueba:</b>	<b>Nombre Historia de Usuario:</b>	Instalar Módulos
<i>HMAST_Web_1-</i>		Multiprocesamiento de Apache2.	
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco			
<b>Descripción de la Prueba:</b> Probar que se instala el MPM <i>worker</i> de Apache2.			
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.			
<b>Entrada / Pasos de ejecución:</b>			
<ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> </ul>			

- Seleccionar el MPM *worker*.

**Resultado Esperado:**

El MPM *worker* se instala y se muestra un mensaje indicando su correcta instalación.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

Código	Caso de Prueba:	Nombre	Historia	de	Usuario:	Instalar	Módulos
<i>HMAST_Web_1-</i>		Multiprocesamiento	de	Apache2.			

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se instala el MPM *event* de Apache2.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Seleccionar el MPM *event*.

**Resultado Esperado:**

El MPM *event* se instala y se muestra un mensaje indicando su correcta instalación.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

Código	Caso de Prueba:	Nombre	Historia	de	Usuario:	Instalar	Módulos
<i>HMAST_Web_1-</i>		Multiprocesamiento.					

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se instala el MPM *threaded* de Apache2.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.

- Seleccionar el MPM *threaded*.

**Resultado Esperado:**

El MPM *threaded* se instala y se muestra un mensaje indicando su correcta instalación.

**Evaluación de la Prueba: Satisfactoria**

### 3.5 Análisis del Rendimiento

Con el fin de demostrar el cumplimiento del objetivo principal del presente trabajo de diploma resulta importante realizar un análisis del rendimiento del módulo Web para HMAST y así verificar que las nuevas funcionalidades implementadas al mismo aumentan su tiempo de respuesta.

La prueba estuvo marcada por las condiciones tecnológicas que se detallan a continuación:

- Computadora Escritorio con GNU/Linux Ubuntu Servidor, procesador 1.40 GHZ, 1GB de RAM, tarjeta Ethernet a 100Mbs, estas propiedades corresponden a la PC sobre la cual se ejecutó la prueba.

La computadora sobre la que se realizó la prueba solo tenía instalado el servidor web Apache2 en su versión 2.22.

La prueba de estrés al servidor Apache2 fue realizada con la herramienta *ApacheJMeter* la cual es una aplicación de escritorio diseñada para cargar el comportamiento funcional de prueba y medir el rendimiento [41].

Bajo el escenario anterior y simulando un nivel de carga de 30, 80 y 150 usuarios conectados que generan 325, 970 y 1960 peticiones simultáneas al servidor web Apache2, se obtuvieron los resultados que se muestran en la siguiente gráfica:



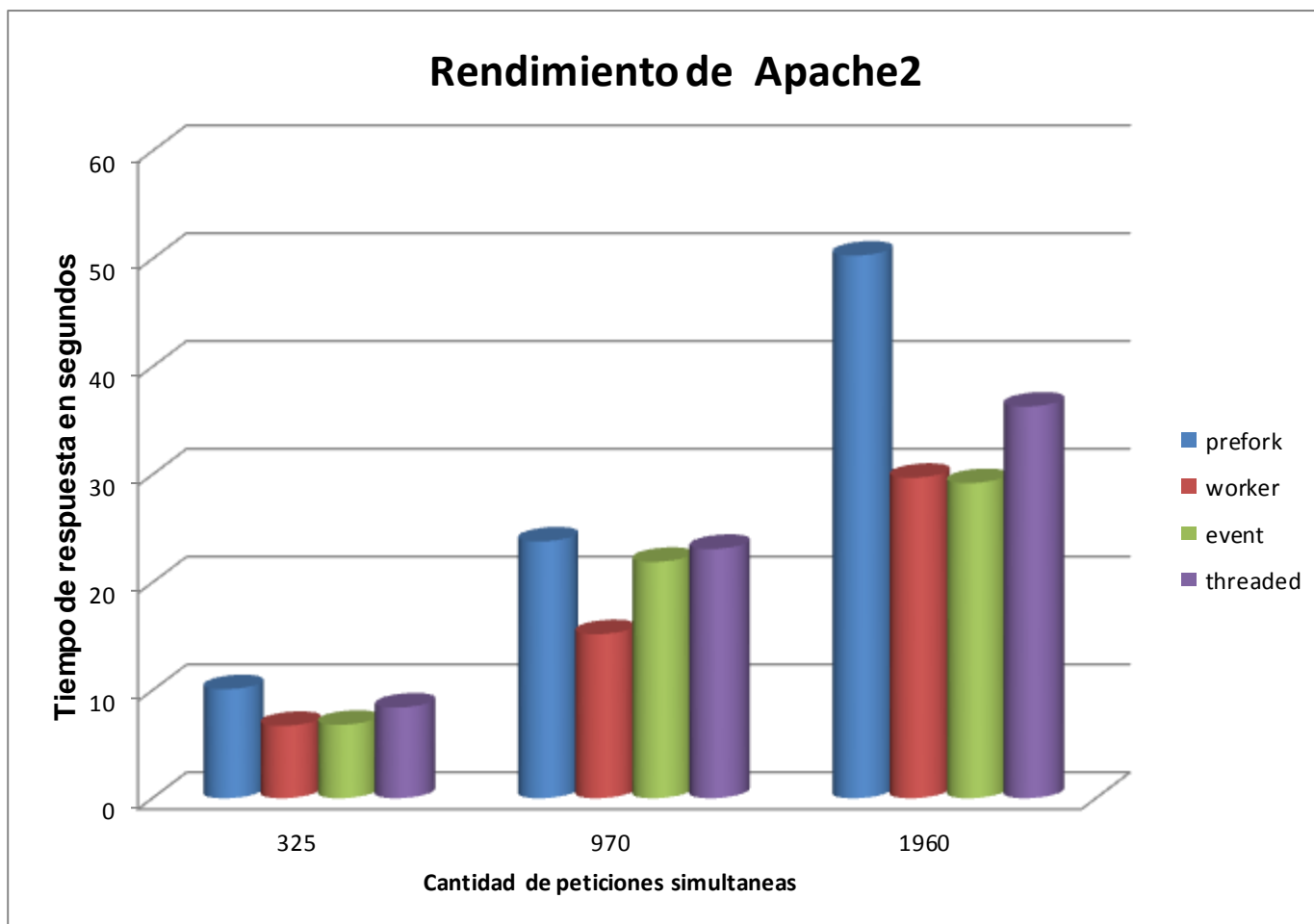


Figura 3.1 Análisis del Rendimiento del servidor Apache2 desde el módulo Web de HMAST.

Los resultados plasmados en la gráfica muestran el tiempo de respuesta que posee el servidor web Apache2 con los distintos MPM instalados y administrados a través del módulo Web de HMAST. Por lo que se puede concluir que la utilización de los MPM *worker*, *event* y *threaded* en el servidor web Apache2 aumentan el rendimiento del mismo posibilitando su utilización en las empresas e instituciones cubanas donde existen poco ancho de banda y condiciones mínimas en cuanto a *hardware*.

### **Conclusiones Parciales**

Con el desarrollo del capítulo, aplicando la metodología de desarrollo de software SXP se generaron las Tareas de Ingeniería y el Plan de Liberación que sirvieron como base para la implementación del módulo diseñado en el capítulo anterior. Se determinó que las pruebas a aplicar son las de aceptación las cuales están definidas por la metodología usada. Las pruebas planificadas permitieron determinar el correcto funcionamiento del módulo implementado así como analizar el rendimiento del mismo concluyendo que los MPM mejoran el tiempo de respuesta del servidor Apache2 desde el módulo Web posibilitando su utilización en las empresas e instituciones cubanas.

### Conclusiones generales

Al terminar el desarrollo del trabajo de diploma se concluye que:

- Se realizó un estudio del estado del arte de las principales características y funcionalidades del servidor web Apache2 específicamente de sus Módulos de Multiprocesamiento, lo que permitió la selección de los MPM *worker*, *event* y *threaded* como los más indicados para su implementación en el módulo Web de HMAST.
- Se desarrollaron funcionalidades como la gestión de los Módulos de Multiprocesamiento *worker*, *event* y *threaded* así como la gestión de directivas globales permitiendo su administración y configuración desde el módulo Web de HMAST. Se incorporó además la capa de Presentación, que posibilitó que el módulo Web contara con una interfaz gráfica permitiendo al usuario configurar Apache2 de una manera más sencilla.
- Se realizaron pruebas a las nuevas funcionalidades implementadas, lo que permitió verificar el correcto funcionamiento de las mismas y el cumplimiento de todos los requerimientos definidos para el módulo Web.
- Se determinó que las nuevas funcionalidades implementadas permiten el aumento del rendimiento del servidor web Apache2 desde el módulo Web de HMAST, posibilitando su utilización en las diferentes empresas e instituciones cubanas donde existe poco ancho de banda y condiciones mínimas en cuanto a *hardware*.

### **Recomendaciones**

Con la realización del presente trabajo de diploma, el módulo Web para HMAST posee las funcionalidades necesarias para administrar y configurar el servidor web Apache2 por lo que se recomienda:

- Continuar profundizando en el estudio del servidor web Apache2.
- Incorporar al módulo Web para HMAST la posibilidad de poder administrar y configurar otros servidores web como Nginx.

**Referencias bibliográficas**

- [1] Fidel Castro. Acto por el aniversario 15 del Palacio Central de Computación. Marzo 2006. [8 Febrero 2012]. Disponible en la Web: <[http://www.cubaminrex.cu/archivo/Presidente/2006/FC\\_070306.htm](http://www.cubaminrex.cu/archivo/Presidente/2006/FC_070306.htm)>.
- [2]. Trabajo\_Diploma\_Integracion\_Sistemas\_PMSWL.pdf. [En línea], Junio 2011[Accedido 13 Febrero 2014].Disponible en: [https://repositorio.geitel.prod.uci.cu/svn/simays/investigaciones/tesis/2010-2011/Abel\\_Jailen/Documento\\_Final/Trabajo\\_Diploma\\_Integracion\\_Sistemas\\_PMSWL.pdf](https://repositorio.geitel.prod.uci.cu/svn/simays/investigaciones/tesis/2010-2011/Abel_Jailen/Documento_Final/Trabajo_Diploma_Integracion_Sistemas_PMSWL.pdf)
- [3]. Castillo Arbelo, Reidiel, y Pablo Soria Acosta. Herramienta para la Migración y Administración de Servidores (HMAS). Junio 2012.
- [4]. Módulos de MultiProcesamiento (MPMs) - Servidor HTTP Apache. [En línea], Marzo 2013 [Accedido 8 Febrero 2014]. Disponible en: <http://apache2.es/2.0.58/mpm.html>
- [5]. mohammed-j-kabir-la-biblia-del-servidor-apache-21.pdf. [En Línea], 2004 [Accedido 13 Febrero 2014]. Disponible en: <http://yexia.files.wordpress.com/2010/09/mohammed-j-kabir-la-biblia-del-servidor-apache-21.pdf>
- [6]. Versión 2.2 de la documentación del Servidor de HTTP Apache - Servidor HTTP Apache. [En línea]. Marzo 2013 [Accedido 13 Febrero 2014]. Disponible en: <https://httpd.apache.org/docs/2.2/es/>
- [7]. Ídem 5
- [8]. ídem 5
- [9]. Ídem 5
- [10] Ídem 5
- [11].Hezigune: LAMP. [En línea]. Enero 2012 [Accedido 8 Febrero 2014]. Disponible en: [http://www.saregune.net/ikasi/hezigune/curso.php?curso=lamp&leccion=lamp\\_apache\\_intro](http://www.saregune.net/ikasi/hezigune/curso.php?curso=lamp&leccion=lamp_apache_intro).
- [12]. Configuración y administración de Apache. [En línea]. 2013 [Accedido 8 Febrero 2014]. Disponible en: <http://dns.bdat.net/documentos/apache/>
- [13].Modelo de la Calidad (ISO/IEC 9126-1:2001, IDT) - NC-ISO-IEC 9126-1.doc -2.Oficina Nacional de Normalizacion.2012.
- [14].Documentación de Servidor Web Apache en castellano/español, en inglés. [En línea]. Marzo 2013 [Accedido 13 Febrero 2014]. Disponible en: <http://apache2.es/>
- [15]. Ídem 6.
- [16]. Ídem 14.
- [17]. Ídem 4.
- [18].Índice de Módulos - Servidor HTTP Apache. [En línea]. Marzo 2013 [Accedido 13 Febrero 2014].

Disponible en: <https://httpd.apache.org/docs/2.2/es/mod/>

[19]. Documentación de Servidor Web Apache en castellano/español, en inglés. [En línea]. Marzo 2013 [Accedido 13 Febrero 2014]. Disponible en: <http://httpd.apache.org/docs/2.0/mod/prefork.html>

[20]. Documentación de Servidor Web Apache en castellano/español, en inglés. [En línea]. Marzo 2013 [Accedido 13 Febrero 2014]. Disponible en: <http://httpd.apache.org/docs/2.0/mod/worker.html>

[21]. Ídem 5

[22]. Módulos de Multiprocesamiento. Primeros Pasos para su configuración. Junio 2013. [En línea]. Disponible en: <http://yoadminsis.blogspot.com/2011/03/instalacion-y-primeros-conceptos-mpm-de.html>

[23]. Documentación de Servidor Web Apache en castellano/español, en inglés, [En línea]. Marzo 2013 [Accedido 13 Febrero 2014]. Disponible en: <http://httpd.apache.org/docs/2.0/mod/perchild.html>

[24] Guía Arquitectura v.2.pdf. 2012 [Accedido 13 Febrero 2014].

[25]. Tesis\_NuriselPalmaPerez\_imprimir.pdf. [En Línea]. Junio 2013 [Accedido 13 Febrero 2014]. Disponible en: [https://repositorio.geitel.prod.uci.cu/svn/simays/investigaciones/tesis/2012-2013/Nurisel/Defensa/Tesis\\_NuriselPalmaPerez\\_imprimir.pdf](https://repositorio.geitel.prod.uci.cu/svn/simays/investigaciones/tesis/2012-2013/Nurisel/Defensa/Tesis_NuriselPalmaPerez_imprimir.pdf).

[26]. Ídem 25

[27]. Bonanata Maximiliano. programacion\_y\_algoritmos.pdf. [En línea]. 2006 [Accedido 23 Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/programacion\\_y\\_algoritmos.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/programacion_y_algoritmos.pdf)

[28]. Holzner Stevens. La\_Biblia\_De\_Java\_2.pdf. [En Línea]. Agosto 2005 [Accedido 23 Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/La\\_Biblia\\_De\\_Java\\_2.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/La_Biblia_De_Java_2.pdf)

[29]. Java How to Program, Seventh Edition.pdf - Java\_How\_to\_Program\_\_7th\_Edition.pdf. [En Línea], 2007 [Accedido 23 Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/Java\\_How\\_to\\_Program\\_\\_7th\\_Edition.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Java_How_to_Program__7th_Edition.pdf)

[30]. Welcome to NetBeans. [En Línea]. 2013. [Accedido 13 Febrero 2014]. Disponible en: <https://netbeans.org/>

[31] Abrams Brad. Framework\_Design\_Guidelines\_2nd\_Edition.pdf. [En Línea]. 2009 [Accedido 23 Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/Framework\\_Design\\_Guidelines\\_2nd\\_Edition.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Framework_Design_Guidelines_2nd_Edition.pdf)

[32]. Professional Java Development with the Spring Framework-Professional\_Java\_Development\_with\_the\_Spring\_Framework.pdf. [En Línea]. 2005 [Accedido el 23 de Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/Professional\\_Java\\_Development\\_with\\_the\\_Spring\\_Framework.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Professional_Java_Development_with_the_Spring_Framework.pdf)

[33]. UML y patrones \_Libro\_.doc - Craig\_Larman\_-\_UML\_y\_Patrones.pdf. [En línea], Septiembre 2007

- [Accedido 23 Mayo 2014]. Disponible en: [https://sunshine.prod.uci.cu/gridfs/sunshine/books/Craig\\_Larman\\_-\\_UML\\_y\\_Patrones.pdf](https://sunshine.prod.uci.cu/gridfs/sunshine/books/Craig_Larman_-_UML_y_Patrones.pdf)
- [34]. Users List. Visual Paradigm for UML. In: Visual Paradigm. [En Línea]. 2012 [Accedido 3 Febrero 2013]. Disponible en: <http://www.visual-paradigm.com/>.
- [35]. Ben Collins-Sussman. Control de versiones con Subversion.pdf, [En Línea], 2009 [Accedido 23 Mayo 2014]. Disponible en: <http://sunshine.prod.uci.cu/gridfs/sunshine/books/SVN.pdf>
- [36]. Tain Domínguez, Yuliette, y Román Miguel Valdivia Genó. Augeas, propuesta tecnológica para la gestión de archivos de configuración de sistemas GNU/Linux. Mayo 2009.
- [37]. S. Pressman Roger In: 6ta edición [En Línea]. [Accedido 8 mayo 2014]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg02670.pdf>.
- [38] PEÑALVER, Gladys Marsi ROMERO. SXP, metodología ágil para proyectos de Software Libre. Habana: Universidad de las Ciencias Informáticas, 2008.
- [39] S. Pressman Roger In: 5ta edición [En Línea]. 2005 [Accedido 11 mayo 2014]. Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
- [40] Diseño con estándares Web - Jeffrey Zeldman. [En Línea]. 2007 [Accedido 11 Mayo 2014]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/Diseno\\_con\\_estandares\\_web.PDF](http://sunshine.prod.uci.cu/gridfs/sunshine/books/Diseno_con_estandares_web.PDF)
- [41] Apache JMeter - Apache JMeterTM. [En Línea]. 2013 [Accedido 11 Mayo 2014]. Disponible en: <http://jmeter.apache.org/>.

## Anexos

The screenshot displays a web management interface with a top navigation bar containing service status boxes: WEB (162.124.201.255), DNS (162.124.201.255), CORREO (correo.uci.cu), FTP (162.124.201.2), DHCP (162.124.201.2), and PROXY (1962.124.10.25). The main content area has three tabs: Módulos, Procesos, and Conexiones. On the left, there are three sidebar buttons: Globales, Servidor Principal, and Host Virtuales. The 'Módulos' tab is active, showing a search bar and a table of modules. A large blue overlay with the text 'Instalando apache2-mpm-worker' is centered over the table. The table lists the following modules and their states:

Módulos	Estado
PhP	<input type="checkbox"/>
Prefork	<input checked="" type="checkbox"/>
Perchild	<input checked="" type="checkbox"/>
Worker	<input type="checkbox"/>

At the bottom right of the overlay, there are two buttons: 'Enviar' (with a checkmark icon) and 'Cancelar' (with an X icon).

Anexo 1: Instalar Módulos de Multiprocesamiento de Apache2.



Anexo 2: Gestionar parámetros en el MPM *worker* de Apache2.

WEB 162.124.201. DNS 162.124.201.255 CORREO correo.ucl.cu FTP 162.124.201.2 DHCP 1962.124.10.25 PROXY

Módulos Procesos Conexiones

Globales

Especificar módulos a usar

Especificar parámetros módulo Threaded

Número de procesos hijos que se crean al iniciar Apache 5

Límite superior del número configurable de procesos 16

Número de hilos en espera Mínimo 250 Máximo 500

Número de hebras creadas por cada proceso hijo 25

Número de hilos creadas por cada proceso hijo 50

Límite de peticiones que un proceso hijo puede atender durante su vida 0

✓Enviar ✕Cancelar

✓Guardar ✕Cancelar

Anexo 3: Gestionar parámetros en el MPM *threaded* de Apache2.

The screenshot shows a web management interface with a top navigation bar containing service status boxes: WEB (162.124.201.255), DNS (162.124.201.255), CORREO (correo.uci.cu), FTP (162.124.201.2), DHCP (162.124.201.2), and PROXY (1962.124.10.25). Below the navigation bar are tabs for 'Módulos', 'Procesos', and 'Conexiones'. A modal dialog titled 'Especificar parámetros módulo event' is open, displaying the following configuration options:

- Número de procesos hijos que se crean al iniciar Apache: 3
- Límite estricto de procesos hijo activos posibles: 32
- Límite superior del número configurable de procesos: 16
- Número de hebras en espera para atender peticiones: Mínimo 75, Máximo 250
- Número de hebras creadas por cada proceso hijo: 25
- Límite superior del número de hebras por proceso hijo: 64
- Límite de peticiones que un proceso hijo puede atender durante su vida: 1000

At the bottom of the dialog are buttons for 'Guardar', 'Cancelar', and 'Cancelar'.

Anexo 4: Gestionar parámetros en el MPM *event* de Apache2.

Anexo 5: Historia de Usuario Especificar parámetros de configuración del Rendimiento de Apache2.

<b>Historia de Usuario</b>	
<b>Número:</b> HMAST_WEB_5	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Nestor A. Delgado Pacheco	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Baja	<b>Puntos Estimados:</b> 1 semana
<b>Riesgo en Desarrollo:</b> Baja	<b>Puntos Reales:</b> 1 semana
<p><b>Descripción:</b> Permitirá mostrar y modificar parámetros de configuración de rendimiento. Este módulo de configuración del rendimiento permitirá configurar distintos parámetros y directivas en Apache2 con el objetivo de acelerar el rendimiento del servidor, atender más peticiones simultáneamente y evitar el colapso de los servicios.</p>	
<p><b>Observaciones:</b> El software permitirá al usuario mostrar y modificar los siguientes parámetros:</p> <ul style="list-style-type: none"> <li>• <b>Activar la resolución de DNS de las direcciones IP de los clientes (<i>HostnameLookups</i>):</b> Esta directiva activa la resolución de DNS de manera que los nombres de host puedan ser guardados en los archivos log. Puede Tomar los valores <i>On</i>, <i>OFF</i> y <i>Double</i>. El valor <i>Double</i> se refiere a hacer una búsqueda de DNS inversa doble. Esto es, después de hacer una búsqueda inversa, se hace una búsqueda normal sobre ese resultado. Al menos una de las direcciones IP en la búsqueda posterior debe equivaler a la dirección IP original. Independientemente de lo que se especifique, cuando <i>mod_access</i> se usa para controlar el acceso por nombre de host, se hará una consulta inversa doble. Esto se hace por seguridad. Tenga en cuenta que el resultado de una búsqueda inversa doble no está disponible generalmente a no ser que especifique <i>HostnameLookups Double</i>. Por ejemplo, si especifica solo <i>HostnameLookups On</i> y se hace una petición a un objeto protegido por restricciones de nombre de host, independientemente de si la consulta inversa doble falla o no, el resultado de la consulta inversa simple se pasará a los CGIs en <i>REMOTE_HOST</i>. El valor por defecto es <i>Off</i> para ahorrar tráfico de red en aquellos sitios web que realmente no necesitan hacer búsquedas inversas dobles. Los sitios web con mucha carga deben usar en esta directiva el valor <i>Off</i>, porque las búsquedas de DNSs</li> </ul>	

pueden consumir una cantidad de tiempo considerable.

- **Buscar ficheros de configuración (*AllowOverrideAll*):** Esta directiva buscaría en los directorios típicos ficheros de configuración *.htaccess*, esto lo que hace es que cada vez que se carga una página web vaya a buscar dicho fichero. Así que es recomendable dejarlo siempre deshabilitado a menos que se desee que un directorio tenga un comportamiento especial. Valor por defecto: deshabilitar.
- **Permitir que se establezcan conexiones HTTP persistentes (*KeepAlive*):** La directiva *KeepAlive*, permite hacer varias peticiones sobre la misma conexión TCP. Esto es particularmente útil cuando se están sirviendo páginas HTML con muchas imágenes. Si *KeepAlive* está en *Off*, entonces se tiene que hacer una conexión TCP para cada imagen, entonces para evitar esto se pondría el *KeepAlive* en *On*. Cuando esta directiva toma valor *On*, se permitirá mostrar y modificar la directiva *KeepAliveTimeout* la cual define el tiempo que se tiene que esperar para la siguiente petición. Se recomienda definir este valor de 2 a 5 segundos, ya que si es demasiado grande, se estaría desaprovechando la posibilidad de ofrecer otro contenido a otro visitante durante la espera.
- **Habilitar Compresión HTTP:** Permitirá habilitar y deshabilitar el módulo *mod\_deflate*, el cual permite disminuir drásticamente el consumo de ancho de banda, pero en su contra consume muchos más recursos, ya que usa *gzip* para comprimir el contenido antes de enviarlo. En Apache2 se puede habilitar la compresión habilitando el módulo *mod\_deflate*.
- **Habilitar Separador de Contenido dinámico y estático:** Permitirá habilitar y deshabilitar los módulos *mod\_proxy* y *rewrite\_module*, los administradores de apache conocen la diferencia en publicar una página HTML que una con PHP, es por eso que mediante los proxys HTTP se le puede decir al apache que por el puerto 80 mande las peticiones de cierto contenido a otro servidor apache y que sólo se encargue de servir imágenes, documentos o páginas web estáticas. Un servidor que ofrece contenido dinámico puede llegar a consumir 20Mb por proceso, en cambio uno de estático como máximo 1Mb. Esto se puede hacer mediante los módulos *mod\_proxy* y *rewrite\_module*.

## Prototipo de interface:

Anexo 6: Tarea de Ingeniería Probar funcionamiento de directivas globales para la configuración de rendimiento en Apache2.

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: HMAST_WEB_5
Nombre Tarea: Probar el funcionamiento de directivas globales para la configuración de rendimiento en Apache2.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2

**Programador Responsable:** Nestor A. Delgado Pacheco

**Descripción:** Se verifica el funcionamiento de directivas globales para la configuración del rendimiento de Apache2. Su configuración se realiza en el fichero de configuración principal `/etc/apache2/apache2.conf`. Las directivas son *HostnameLookups*, *AllowOverride All*, *KeepAlive*.

Anexo 7: Casos de Prueba de Aceptación del módulo Web de HMAST.

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>StartServers</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>StartServers</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM	

*worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

**Código Caso de Prueba:**

*HMAST\_Web\_2-*

**Nombre Historia de Usuario:** Gestionar parámetros en el MPM *worker* de Apache2.

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *MaxClients*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

**Código Caso de Prueba:**

*HMAST\_Web\_2-*

**Nombre Historia de Usuario:** Gestionar parámetros en el MPM *worker* de Apache2.

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *MaxClients*.



**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
--	---

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *MinSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
--	---

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *MinSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
--	---

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *MaxSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *worker* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *worker*.
- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
--	---

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>MaxSpareThreads</i> .
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.
<b>Evaluación de la Prueba: Satisfactoria</b>

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>ThreadsPerChild</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.

<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>ThreadsPerChild</i> .
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.
<b>Evaluación de la Prueba:</b> Satisfactoria

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>ThreadLimit</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>ThreadLimit</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>	
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>MaxRequestsPerChild</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_2-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>worker</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>MaxRequestsPerChild</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>worker</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>worker</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>	
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>StartServers</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>event</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>event</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM event de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>StartServers</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>event</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>event</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>	
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>MaxClients</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>event</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>event</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>	
<b>Resultado Esperado:</b>	

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

#### Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *MaxClients*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

#### Entrada / Pasos de ejecución:

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

#### Resultado Esperado:

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

#### Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *MinSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

#### Entrada / Pasos de ejecución:

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.



- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

**Código Caso de Prueba:**

*HMAST\_Web\_3-*

**Nombre Historia de Usuario:** Gestionar parámetros en el MPM *event* de Apache2.

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *MinSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

**Código Caso de Prueba:**

*HMAST\_Web\_3-*

**Nombre Historia de Usuario:** Gestionar parámetros en el MPM *event* de Apache2.

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *MaxSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.

- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *MaxSpareThreads*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *ThreadsPerChild*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.

- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *ThreadsPerChild*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se muestra el valor de la directiva *ThreadLimit*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM

*event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Probar que se modifica el valor de la directiva *ThreadLimit*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2. Tener instalado el MPM *event* de Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Abrir la configuración de los MPM.
- Abrir la configuración del MPM *event*.
- Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.

**Resultado Esperado:**

El valor de la directiva en el fichero de configuración es el entrado por el usuario.

**Evaluación de la Prueba: Satisfactoria**

Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>MaxRequestsPerChild</i> .
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>event</i> de Apache2.
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>event</i>.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.
<b>Evaluación de la Prueba: Satisfactoria</b>

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_3-</i>	<b>Nombre Historia de Usuario:</b> Gestionar parámetros en el MPM <i>event</i> de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>MaxRequestsPerChild</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2. Tener instalado el MPM <i>event</i> de Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Abrir la configuración de los MPM.</li> <li>• Abrir la configuración del MPM <i>event</i>.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>	
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	

<b>Descripción de la Prueba:</b> Probar que se muestra el valor de la directiva <i>KeepAlive</i> .
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Verificar que el valor que tiene la directiva en el fichero es la misma que la mostrada en la interfaz.</li> </ul>
<b>Resultado Esperado:</b> Coincide el valor de la interfaz con el del fichero <i>apache2.conf</i> del módulo Web.
<b>Evaluación de la Prueba: Satisfactoria</b>

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Probar que se modifica el valor de la directiva <i>KeepAlive</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Verificar que la directiva toma el valor entrado por el usuario en el fichero de configuración.</li> </ul>	
<b>Resultado Esperado:</b> El valor de la directiva en el fichero de configuración es el entrado por el usuario.	
<b>Evaluación de la Prueba: Satisfactoria</b>	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Permitirá mostrar el estado del módulo <i>mod_deflate</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> </ul>	

- Verificar que el estado que tiene el módulo en el fichero es la misma que la mostrada en la interfaz.

**Resultado Esperado:**

Coincide el valor de la interfaz con el del fichero *apache2.conf* del módulo Web.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Permitirá habilitar módulo *mod\_deflate*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Verificar que el módulo *mod\_deflate* se encuentre habilitado.

**Resultado Esperado:**

El estado del módulo *mod\_deflate* es *true*.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Permitirá deshabilitar módulo *mod\_deflate*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Verificar que el módulo *mod\_deflate* se encuentre deshabilitado.

**Resultado Esperado:**

El estado del módulo *mod\_deflate* es *false*.

## Evaluación de la Prueba: Satisfactoria

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Permitirá habilitar módulo *mod\_proxy*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Verificar que el módulo *mod\_proxy* se encuentre habilitado.

**Resultado Esperado:**

El estado del módulo *mod\_proxy* es *true*.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación

<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
--	--

**Nombre de la persona que realiza la prueba:** Nestor A. Delgado Pacheco

**Descripción de la Prueba:** Permitirá deshabilitar módulo *mod\_proxy*.

**Condiciones de Ejecución:** Estar conectado a un servidor Web Apache2.

**Entrada / Pasos de ejecución:**

- Abrir el módulo de Web.
- Verificar que el módulo *mod\_proxy* se encuentre deshabilitado.

**Resultado Esperado:**

El estado del módulo *mod\_proxy* es *false*.

**Evaluación de la Prueba: Satisfactoria**

## Caso de Prueba de Aceptación



<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Permitirá habilitar módulo <i>rewrite_module</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Verificar que el módulo <i>rewrite_module</i> se encuentre habilitado.</li> </ul>	
<b>Resultado Esperado:</b> El estado del módulo <i>rewrite_module</i> es <i>true</i> .	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> <i>HMAST_Web_4-</i>	<b>Nombre Historia de Usuario:</b> Especificar parámetros de configuración del Rendimiento de Apache2.
<b>Nombre de la persona que realiza la prueba:</b> Nestor A. Delgado Pacheco	
<b>Descripción de la Prueba:</b> Permitirá deshabilitar módulo <i>rewrite_module</i> .	
<b>Condiciones de Ejecución:</b> Estar conectado a un servidor Web Apache2.	
<b>Entrada / Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• Abrir el módulo de Web.</li> <li>• Verificar que el módulo <i>rewrite_module</i> se encuentre deshabilitado.</li> </ul>	
<b>Resultado Esperado:</b> El estado del módulo <i>rewrite_module</i> es <i>false</i> .	
<b>Evaluación de la Prueba:</b> Satisfactoria	

### Glosario de términos

**Directivas:** son variables almacenadas en el archivo de texto de configuración para alterar y controlar el funcionamiento de Apache en tiempo de ejecución según sus valores y después de haber reiniciado el proceso servidor.

**Hilo:** es un proceso totalmente aislado y es el responsable de ejecutar el código contenido en el espacio de direcciones del proceso

**Hebrado:** Término utilizado por la bibliografía del servidor web Apache2 para referirse a un servidor multiproceso.

**Hebras:** Varios hilos de ejecución.

**Host virtual basados en nombre:** se refiere a la asignación de varios hosts virtuales a una única combinación direcciónIP:puerto, por tanto esta dirección IP escuchando por ese puerto tendría asociado un único directorio raíz.

**Host virtual basado en IP:** se refiere a la asignación de un solo host virtual a una única combinación direcciónIP:puerto, por tanto esta dirección IP escuchando por ese puerto tendría asociado un único directorio raíz.

**Módulos de Multiprocesamiento (MPM):** son los encargados del manejo de peticiones en Apache2, responsables de conectar con los puertos de red de la máquina, aceptar las peticiones y generar los procesos hijos que se encargan de servirlos.

**Modularización:** Dividir un problema en segmentos simples.

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

**Proceso:** es una instancia u ocurrencia de un programa en ejecución, es propietario de una serie de recursos como un espacio de direcciones en memoria, ficheros, hilos, entre otros. Para que un proceso sea capaz de hacer algo debe ser propietario de al menos un hilo (*thread*), aunque puede contener varios.

**Socket:** es un punto o medio de comunicación entre dos aplicaciones que permite que un proceso emita o reciba información de otro proceso estando los dos en distintas máquinas.

**Servidor web:** es un programa que procesa cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.