



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas

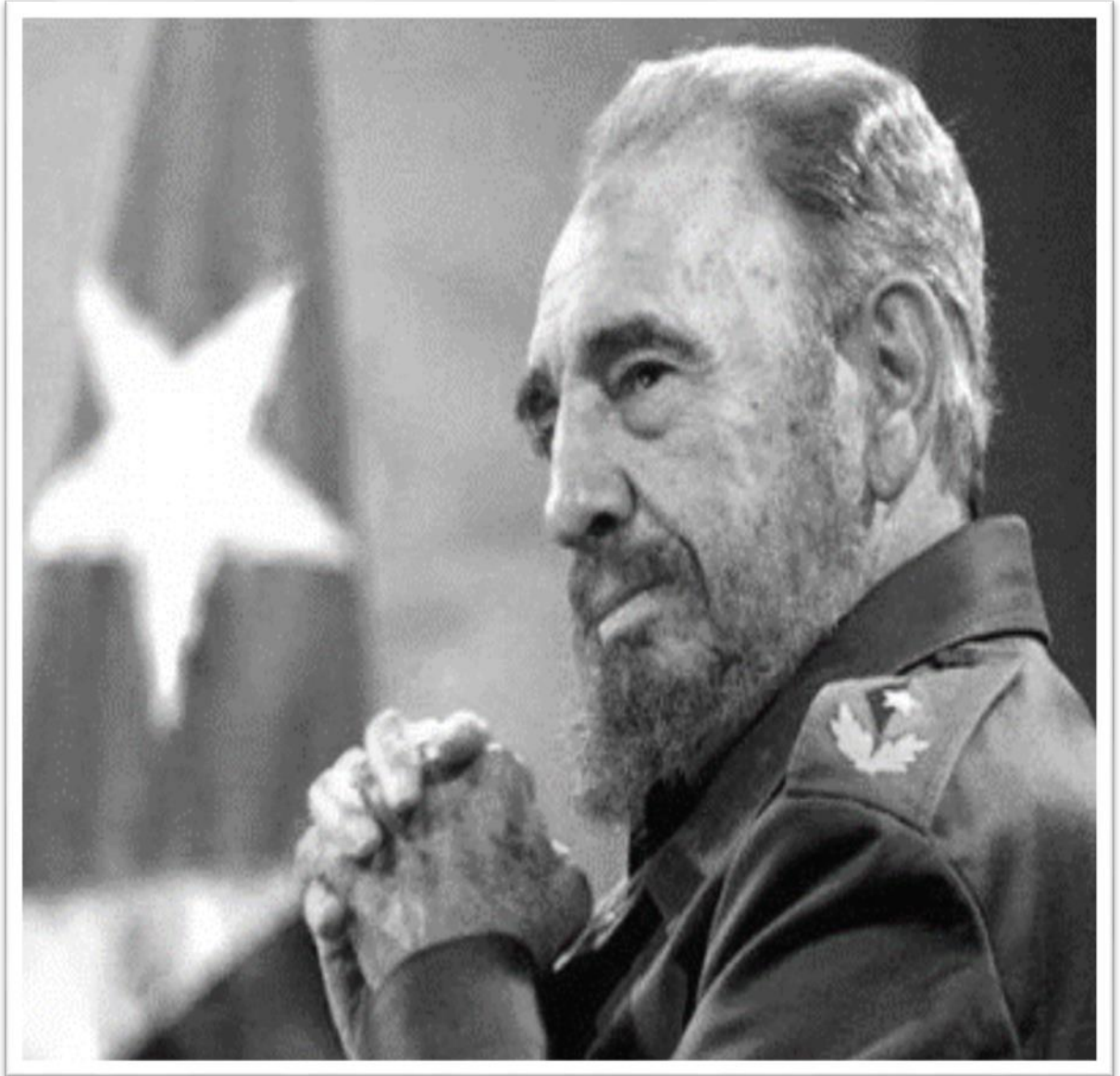
Sistema de gestión, administración y monitoreo de servidores web
Apache para el Departamento de Componentes del Centro de
Telemática

Autor: Emilio Gort Caballero

Tutor: Ing. Osmar Capote Vázquez

La Habana, junio de 2018

“Año 60 de la Revolución”



"Mientras más difíciles sean las circunstancias, más alta tiene que ser nuestra moral, más elevado tiene que ser nuestro espíritu, más sólida nuestra firmeza"

Fidel Castro Ruz

Agradecimientos

A mis padres y mi hermana por brindarme lo mejor de sí y por sembrar en mi la fe del triunfo.

A mi tutor Osmar por guiar con su experiencia cada paso que ejecuté en la realización exitosa de este trabajo.

A todos mis profesores que a lo largo de esta hermosa carrera contribuyeron a mi formación profesional.

A todos mis compañeros por haber dedicado un espacio de su tiempo y que de una forma u otra aportaron su granito de arena en este trabajo.

A la Revolución y a su invicto y eterno Comandante en Jefe Fidel Castro Ruz, por haberme dado la oportunidad de estudiar esta profesión y haber puesto en mi la confianza de convertirme en un hombre de ciencias.

Dedicatoria

A mis padres por haberme dado la luz de la vida y ser el motor impulsor para realizar este trabajo.

A mi hermana por su apoyo incondicional y por servirme de estímulo y soporte para llegar hasta el final.

A mi familia, en especial a mi abuela materna, por sus consejos, por inculcarme los mejores valores humanos, por su paciencia y por confiar en mí.

A mi novia por su ternura, comprensión y por estar todo el tiempo pendiente de mí.

A todos ellos, por haber servido como fuente de inspiración para hacer de esta investigación un sueño realizado.

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los_____días del mes de junio del año_____.

Emilio Gort Caballero

Autor

Ing. Osmar Capote Vázquez

Tutor

Resumen

El Departamento de Componentes perteneciente al Centro de Telemática de la Universidad de las Ciencias Informáticas, está compuesto por un grupo de especialistas, los cuales se preparan y se actualizan a diario con aplicaciones a clientes, servicios de bases de datos, servicios de monitoreo y servicios web. En la actualidad, en el departamento, las configuraciones de los servicios web Apache no se realizan de forma gráfica sino a través de una terminal a base de comandos. Esto trae consigo que este proceso se vuelva lento y engorroso para los especialistas y puede conllevar a la ocurrencia de errores humanos durante el proceso. La presente investigación tiene como objetivo la descripción de una propuesta que facilite el proceso de gestión, administración y monitorización de los servicios web Apache que se realizan en dicho departamento. Para guiar el desarrollo de la de la misma se utilizó Programación Extrema como metodología de desarrollo de software, la cual proporcionó los artefactos para obtener el sistema deseado, así como el uso de diferentes herramientas informáticas empleadas en el desarrollo del mismo. Como resultado se obtuvo un sistema capaz de gestionar, administrar y monitorizar los servicios web Apache de forma centralizada para el Departamento de Componentes del Centro de Telemática de la facultad 2.

Palabras claves: Apache, servicios web, sistema.

Abstract

The Department of Components belonging to the Center for Telematics of the University of Computer Science, is composed of a group of specialists, which are prepared and updated daily with applications to clients, database services, monitoring services and services Web. Currently, in the department, the configurations of Apache web services are not performed graphically but through a command based terminal. This means that this process becomes slow and cumbersome for specialists and can lead to the occurrence of human errors during the process. The objective of this research is the description of a proposal that improves the management, administration and monitoring processes of the Apache web services that are carried out in said department. To guide the development of the same, Extreme Programming was used as software development methodology, which provided the artifacts to obtain the desired system, as well as the use of different computer tools used in the development of the same. As a result, a system capable of managing, administering and monitoring the Apache web services was obtained in a centralized manner for the Components Department of the Telematics Center of faculty 2.

Keywords: Apache, web services, system.

Índice general

Introducción	1
Capítulo 1: fundamentación teórica.....	8
Introducción.....	8
1.1 Conceptos asociados	8
1.2 Análisis del estado del arte.....	9
1.2.1 Internacional	9
1.2.2 Universidad de las Ciencias Informáticas	14
1.3 Metodología de Desarrollo de Software.....	15
1.3.1 Programación Extrema (XP)	16
1.4 Herramientas y lenguajes informáticos	16
1.4.1 Lenguaje de programación del lado del servidor	16
1.4.3 Framework de desarrollo	18
Conclusiones del capítulo.....	21
Capítulo 2: Análisis y diseño del sistema.....	22
Introducción.....	22
2.1 proceso de administración de servicios web apache	22
2.2 características de la propuesta de solución	23
2.3 roles y funcionalidades	24
2.4 fase de exploración	25
2.4.1 historias de usuario.....	25
2.5 fase de planificación	32
2.5.1 estimación del esfuerzo	32
2.5.2 plan de iteraciones.....	33
2.5.3 plan de entrega	35
2.6 fase de diseño	35
2.6.1 patrones de arquitectura.....	35
2.7 patrones de diseño.....	38
2.8 tarjetas clase responsabilidad colaboración (crc)	39
conclusiones del capítulo	41
Capítulo 3: Implementación y pruebas	42
Introducción.....	42

3.1 Implementación	42
3.1.2 Tareas de Ingeniería	44
3.2.1 Pruebas unitarias	47
3.2.2 Pruebas de aceptación	48
3.4 Conclusiones del capítulo	57
Conclusiones	58
Recomendaciones	59
Glosario de términos	60
Referencias bibliográficas	62
Bibliografía	66

Índice de figuras

Figura 1 Interfaz de Applications Manager	10
Figura 2 Interfaz del Webmin	11
Figura 3 Interfaz de Apache GUI	12
Figura 4 Interfaz de cPanel	13
Figura 5 Interfaz de la Plataforma Nubity	13
Figura 6 Interfaz de HMAST	14
Figura 7 Proceso del negocio actual	22
Figura 8 Propuesta de solución del sistema	24
Figura 9 arquitectura Cliente-Servidor	36
Figura 10 MTV	38
Figura 11 Importaciones	43
Figura 12 CapWords	43
Figura 13 lower_case_with_underscores	43
Figura 14 Resultado de la ejecución de las pruebas unitarias utilizando el módulo PyUnit	48
Figura 15 Resultados de la ejecución de pruebas de aceptación	56

Índice de tablas

Tabla 1 Ficha de proceso	22
Tabla 2 Historia de usuario "Autenticar usuario"	26
Tabla 3 Historia de usuario "Gestionar máquinas servidoras"	27
Tabla 4 Historia de usuario "Establecer conexión por SSH a un servidor remoto"	27
Tabla 5 Historia de usuario "Instalar servicio Apache"	28
Tabla 6 Historia de usuario "Cambiar estado del servicio Apache"	29
tabla 7 Historia de usuario "Desinstalar servicio Apache"	29
tabla 8 Historia de usuario "Gestionar Hosts Virtuales"	30
tabla 9 Historia de usuario "Mostrar Logs"	30
tabla 10 Historia de usuario "Monitorear servicio Apache"	31
Tabla 11 Estimación de esfuerzo	32
Tabla 12 Plan de duración de las iteraciones	34
Tabla 13 Plan de Entrega	35
Tabla 14 Tarjeta CRC No.1	40
Tabla 15 Tarjeta CRC No.2	40
Tabla 16 Tarjeta CRC No.3	40
Tabla 17 Tarjeta CRC No.4	41
Tabla 18 Tarea de ingeniería "Autenticar usuario"	44
Tabla 19 Tarea de ingeniería "Establecer conexión por SSH a un servidor remoto"	44
Tabla 20 Tarea de ingeniería "Instalar servicio Apache"	44
Tabla 21 Tarea de ingeniería "Cambiar estado del servicio Apache"	45
Tabla 22 Tarea de ingeniería "Desinstalar servicio Apache"	45
Tabla 23 Tarea de ingeniería "Mostrar Logs"	46

Tabla 24 Tarea de ingeniería “Monitorear servicio Apache”	46
Tabla 25 Tarea de ingeniería “Adicionar hosts virtuales”	46
Tabla 26 Tarea de ingeniería “Eliminar hosts virtuales”	47
Tabla 27 Prueba de aceptación “Autenticar usuario”	48
Tabla 28 Prueba de aceptación “Establecer conexión por SSH a un servidor remoto”	48
Tabla 29 Prueba de aceptación “Instalar servicio Apache”	49
Tabla 30 Prueba de aceptación “Cambiar estado del servicio Apache”	49
Tabla 31 Prueba de aceptación “Eliminar Host Virtual”	50
Tabla 32 Prueba de aceptación “Habilitar y deshabilitar Host Virtual”	51
Tabla 33 Prueba de aceptación “Desinstalar servicio Apache”	51
Tabla 34 Prueba de aceptación “Restringir acceso al servicio Apache”	52
Tabla 35 Prueba de aceptación “Modificar puertos para las conexiones en Apache”	52
Tabla 36 Prueba de aceptación “Mostrar Logs”	53
Tabla 37 Prueba de aceptación “Monitorear servicio Apache”	53
Tabla 38 Prueba de aceptación “Adicionar usuario”	54
Tabla 39 Prueba de aceptación “Eliminar usuario”	54
Tabla 40 Prueba de aceptación “Editar Host Virtual”	55
Tabla 41 “No conformidades identificadas por iteraciones”	56

Introducción

En los últimos años, las tecnologías de la información y las comunicaciones (TIC) ocupan un lugar cimero en el desarrollo de la sociedad y la economía. Este concepto nace con la convergencia tecnológica de la electrónica, el software y la infraestructura de las telecomunicaciones. El avance que se registra en el campo de dichas tecnologías, ha permitido llevar la globalidad al mundo de la comunicación y facilitar la interconexión entre las personas e instituciones a nivel mundial (Damian, 2017). No se puede negar que toda la vida política, económica, social y cultural del globo terráqueo, está íntimamente ligada a los impactos de las TIC.

La dimensión social de las TIC se refleja en la fuerza e influencia que tiene en los diferentes ámbitos y en las nuevas estructuras sociales. En la actualidad constituyen unos de los factores de innovación más importantes en el campo de la economía y dirección de empresas, pues contribuyen al crecimiento y aumento de la productividad de la misma (Consuelo, 2012). Por otra parte, las TIC proveen de herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos sociales actuales. Herramientas que realizan de forma rápida y precisa muchos procesos y también han transformado la manera de comunicarse y como acceder e interactuar con la información mediante el uso de servidores.

Los servidores han evolucionado, principalmente sus capacidades de procesamiento y almacenamiento. Estos sistemas se ocupan de gestionar el tráfico en la red y brindar una gran cantidad de servicios a los clientes, como acceso a ficheros e Internet, correo electrónico, entre otros. Dentro de los diferentes tipos de servidores se encuentra el servicio web, el cual tiene como rol fundamental estandarizar la comunicación entre distintas plataformas y lenguajes de programación.

Un servicio web se mantiene a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Se encarga de contestar a estas peticiones de forma adecuada y entregar como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Proporciona los recursos mediante el protocolo HTTP (*Hypertext Transfer Protocol*) o el protocolo HTTPS (*Hypertext Transport Protocol Secure*), la versión segura, cifrada y autenticada de HTTP (josegregoriob, 2011). Existen numerosos servicios web dentro de los que se encuentran: Microsoft IIS, Sun Java System Web Server, Nginx, Lighttp, LiteSpeed, Roxen, Google Servers, y Apache, el cual ofrece instalaciones sencillas para sitios pequeños y de código abierto para una multitud de plataformas y de sistemas operativos.

El servicio web Apache surge en el año 1995 y actualmente se desarrolla dentro del proyecto *HTTP Server de Apache Software Foundation* (kabir, 2004). Este servidor se caracteriza por poseer un sistema de módulos dinámicos, los cuales pueden ser habilitados o deshabilitados mediante la ejecución de

comandos o reinicio del servidor. Esto posibilita que el servidor sea adaptado a diferentes entornos y necesidades. Permite que estén en marcha aquellas funcionalidades necesarias y optimizar así su rendimiento.

Apache trae por defecto un sitio web que es la base sobre el cual se construyen los hosts virtuales, el cual es llamado servidor principal. El servidor soporta los hosts virtuales basados en IP (*Internet Protocol*) y los basados en nombre. Una PC (computadora) servidora puede tener asignada una o varias direcciones IP y cada dirección puede escuchar por una lista de puertos específicos y además por todos los puertos. A cada combinación dirección IP y puerto se puede asignar un host virtual basado en IP o una lista de host virtuales basados en nombre (Martínez, 2013).

Apache introduce los módulos de multiprocesamiento (MPM) en el manejo de peticiones, los cuales 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 (Apache, 2017). Cuando llega una petición al servidor se realiza una búsqueda del host virtual que debe atenderla. En caso de que no exista coincidencias entre la dirección IP o el puerto de la petición con los asignados a los hosts virtuales, la petición será atendida por el servidor principal.

Apache cuenta además con otras características como soporte del último protocolo HTTP 1.1.2. Configuración simple y potente a través de ficheros. Soporte de autenticación HTTP, soporte de scripts PHP, soporte de servlets de Java, soporte de SSI (Server Side Includes) y de SSL (*Secured Socket Layer*). Negociación avanzada de los contenidos que gestiona el servidor. Reescritura y chequeo avanzado de sintaxis de la URL (Kabir, 2004).

Cuba ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC. Lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia un desarrollo sostenible. Para ello, el país se ha propuesto alcanzar una dependencia tecnológica, como uno de los pilares de desarrollo del programa de informatización de la sociedad cubana. Para llevar a cabo dicha informatización se les ha delegado la tarea a instituciones desarrolladoras de software. Entre las más prestigiosas se encuentra la Universidad de Ciencias Informáticas (UCI), surgida en el fragor de la Batalla de Ideas y obra del pensamiento visionario del Comandante en Jefe Fidel Castro. La institución tiene dentro de sus objetivos desarrollar la industria del software para contribuir al desarrollo económico (Falcón Márquez, 2016).

La universidad cuenta con diversos centros de desarrollo. Entre ellos se encuentra el Centro de Telemática (TLM), donde se desarrollan sistemas y servicios informáticos integrales para las Telecomunicaciones y la Seguridad Informática. Además, integra los procesos docentes, productivos e

investigativos con alto nivel, contando con un personal especializado en dichas áreas (Hernández, 2018). El centro cuenta entre sus instalaciones con el Departamento de Componentes, en el cual sus trabajadores se preparan y se actualizan a diario con las aplicaciones a clientes, servicios web, servicios de base de datos, servicios de monitoreo, entre otros. Realizan pruebas en el departamento para instruirse, aprender cómo funcionan esos servicios, explotarlos en el área de pruebas y posteriormente se realizan consultorías a las empresas, donde ponen en práctica el conocimiento adquirido.

En la actualidad, en el Departamento de Componentes del Centro de Telemática, las configuraciones de los servicios web Apache no se realizan de forma gráfica sino a través de una terminal a base de comandos. Esto trae consigo que este proceso se vuelva lento y engorroso para los especialistas y puede conllevar a la ocurrencia de errores humanos durante el proceso. Además, existe la necesidad de obtener información de los recursos del servidor que son utilizados por el servicio Apache y detectar errores en la configuración del mismo para verificar si dicho servicio se está ejecutando correctamente. Algunos softwares de forma independiente realizan una u otra función, pero ninguno cumple con las necesidades específicas del área.

Para dar solución a la problemática existente descrita anteriormente se plantea como **problema a resolver**: ¿Cómo gestionar, administrar y monitorear los servidores web Apache de manera centralizada en el Departamento de Componentes del Centro de Telemática?

Como **Objeto de estudio** se define: El proceso de gestión, administración y monitoreo de servidores web Apache.

Teniendo en cuenta el problema planteado se define como **objetivo general**: Desarrollar un sistema informático que permita gestionar, administrar y monitorear los servidores web Apache en el Departamento de Componentes del Centro Telemática.

Para dar cumplimiento al objetivo general se precisan los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación para sustentar el desarrollo de un sistema de gestión, administración y monitorización de servicios web Apache.
2. Definir las funcionalidades que debe cumplir el sistema para la gestión, administración y monitorización de servicios web apache.
3. Implementar un sistema informático para gestionar, administrar y monitorear los servicios web que cumpla con los requerimientos definidos.
4. Aplicar pruebas al sistema resultante para garantizar la calidad y fiabilidad del producto.

Campo de acción: La gestión, administración y monitoreo de servidores web Apache en el Departamento de Componentes del Centro de TLM.

Para presidir el desarrollo de la presente investigación se definen las siguientes **preguntas científicas**:

- ¿Cuáles son las posiciones teóricas en cuanto a los sistemas para la gestión, administración y monitoreo de los servidores web Apache?
- ¿Cuál es el estado real de desarrollo alcanzado en la informatización de las áreas involucradas en el flujo y manejo de información de los servicios de Apache?
- ¿Cómo diseñar un sistema para la gestión, administración y monitoreo que se ajuste a las características de los servidores web Apache?

Para dar cumplimiento a lo anteriormente expuesto se plantean las siguientes **tareas de investigación**:

1. Análisis de sistemas homólogos existentes para conocer aspectos regulares en el diseño del sistema enfocados a los servicios web Apache.
2. Caracterización de las herramientas informáticas y metodología de desarrollo de software para realizar la implementación del sistema.
3. Identificación de las principales funcionalidades del sistema para la posterior implementación del mismo.
4. Análisis sobre los diferentes tipos de prueba que propone la metodología de desarrollo de software para validar el correcto funcionamiento del sistema.

Método científico de investigación

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados (González, 2012).

Métodos de investigación teóricos

Permiten estudiar las características del objeto de investigación que no son observables directamente. Facilitan la construcción de modelos. Crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad. Se apoyan en el proceso de análisis y síntesis (González, 2012).

Analítico-Sintético (González, 2012)

Análisis: Permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio.

Síntesis: Establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas.

Utilizado para analizar la información obtenida en el proceso de investigación en el Departamento de Componentes del Centro de TLM de la facultad 2, lo que posibilitó la identificación de aspectos empleados para el desarrollo de la presente investigación. Además, para el análisis de la información referente dentro y fuera del país acerca de los servicios web Apache, sintetizando aquellos aspectos más significativos al respecto, así como arribar a conclusiones.

Inductivo-Deductivo (González, 2012)

Inducción: Es una forma de razonamiento a través del cual se pasa de un conocimiento de cosas particulares a un conocimiento más general que va a reflejar lo que hay de común en esos fenómenos individuales.

Deducción: Es un procedimiento que permite a partir de conocimientos generales inferir casos particulares por un razonamiento lógico.

Utilizado para analizar el comportamiento y las características que poseen las herramientas informáticas para la gestión, administración y monitorización de los servicios web Apache. Reunir elementos para obtener una conclusión general a partir de los datos particulares del por qué es necesario implementar un sistema que cumpla con los requerimientos establecidos.

Métodos empíricos

Describen y explican las características fenomenológicas del objeto. Representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional (González, 2012).

Modelado

La modelación es el método mediante el cual se crean abstracciones con el objetivo de explicar la realidad. El modelo como sustituto del objeto de investigación es semejante a él, existiendo una correspondencia objetiva entre el modelo y el objeto, siendo el investigador quien elabora dicho modelo.

Se empleó para reflejar el proceso de gestión, administración y monitorización de los servicios web Apache actualmente dentro del Departamento de Componentes, mediante un diagrama de proceso de negocio auxiliar, para un mejor entendimiento. Además, para modelar las relaciones y características de la solución propuesta. También facilitó el diseño de clases necesario para la implementación del sistema.

Observación

La observación científica es la percepción planificada dirigida a un fin y relativamente prolongada de un hecho o fenómeno. Es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado.

Se utilizó para la búsqueda de información referente a los sistemas desarrollados, asociados a la gestión, administración y monitoreo de los servicios web Apache y analizar los resultados obtenidos sobre el comportamiento de los mismos.

Entrevista

La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado. Puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación.

Se utilizó como fuente de recopilación de información, aplicada al cliente a y los trabajadores del Departamento de Componentes para entender cómo se desarrolla el negocio, los procesos que se llevan a cabo para la gestión, administración y monitorización de los servidores web apache. Se realizan entrevistas con el tutor para verificar el trabajo realizado. Las preguntas realizadas en la entrevista pueden consultarse en el **Anexo 1**.

El trabajo de diploma consta de tres capítulos, las conclusiones generales, la bibliografía general utilizada y el glosario de términos donde se explican los términos de difícil comprensión que se han utilizado para la elaboración de esta investigación. La estructura de los capítulos se define de la siguiente forma:

Capítulo 1. “Fundamentación teórica”

En este capítulo se realiza el análisis del estado del arte de sistemas homólogos que realizan la administración, gestión y monitoreo de servicios web Apache. Se exponen los principales conceptos asociados que ayudarán a familiarizarse con el entorno en el que se manifiesta la investigación. Además, se selecciona la metodología de desarrollo de software, las herramientas y lenguajes informáticos para el desarrollo del sistema.

Capítulo 2. “Análisis y diseño del sistema”

En este capítulo se define una propuesta de solución para resolver el problema planteado. Se determinan los roles y las funcionalidades a implementar en el sistema. Se plantean las fases de exploración,

planificación y diseño. Además de realizar una descripción del flujo del negocio y las características de la propuesta de solución a través del modelado de proceso de negocio.

Capítulo 3. “Implementación y pruebas”

En este capítulo se realiza la implementación del sistema a partir de las tareas de ingeniería generadas por la metodología XP (*eXtreme Programming* por sus siglas en inglés). Se detallan los estándares de codificación que se tuvieron en cuenta para desarrollar el sistema. Se diseñan y realizan los casos de pruebas para verificar el comportamiento del sistema y el cumplimiento de los objetivos.

Capítulo 1: Fundamentación teórica

Introducción

En este capítulo se definen los elementos teóricos que sustentan la investigación. Se hace referencia a los conceptos asociados a la administración de servidores web Apache. Se realiza un estudio del estado del arte de sistemas informáticos para la gestión, administración y monitoreo de servidores web Apache a nivel nacional e internacional. Además, se determinan las herramientas informáticas y el tipo de metodología a utilizar para el desarrollo del software.

1.1 Conceptos asociados

Servicio

Un servicio es el conjunto de actividades que se relacionan entre sí y de actitudes que se diseñan para satisfacer las necesidades de los usuarios. Un servicio es también el resultado de un proceso, es decir, el resultado de llevar a cabo necesariamente al menos una actividad en la interfaz entre el proveedor y el cliente y generalmente es intangible. Es cualquier beneficio que una parte ofrece a otra y su producción puede estar vinculada o no con un producto físico (Pérez Ríos, 2014).

Servicio Web

El servicio *World Wide Web* o simplemente web, se podría definir como un amplio sistema multimedia de acceso a información distribuida por toda la red en forma de documentos hipertextuales. La información reside en forma de páginas web en ordenadores que se denominan servidores web. Es una aplicación software identificada por un URI (*Uniform Resource Identifier*) cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML (Lenguaje-Marcado-Extensible) intercambiados mediante protocolos de internet (Monsalve, 2015).

Servidor web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor. Realiza conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente. Genera o cede una respuesta en cualquier lenguaje o aplicación del lado del cliente (Gregorio, 2011).

Servicio web Apache

El servicio web Apache HTTP es un servicio de código abierto para una multitud de plataformas y de sistemas operativos. Permite la creación de páginas y servicios web. Es altamente configurable, admite

bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Adaptado a los nuevos protocolos, y cuya implementación se realiza de forma colaborativa (JM Pantoja Blyde, 2013) .

Sistema informático

Un sistema informático, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano que permite almacenar y procesar información (Miranda , 2012).

1.2 Análisis del estado del arte

En el mundo existen diferentes sistemas informáticos homólogos que responden a las mismas necesidades que justifican el desarrollo de la presente investigación: administrar, gestionar y monitorear los servicios web Apache. Para ello se realizó un estudio analítico de los mismos para saber si sería posible utilizar uno de ellos o realizar el desarrollo de un nuevo sistema informático.

1.2.1 Internacional

Applications Manager:

Applications Manager, como se muestra en la figura 1, es un software de monitorización de aplicaciones que ayuda a asegurar la máxima disponibilidad y rendimiento de sus aplicaciones de negocio. Se trata de un sistema de APM (*Application Performance Management*) o Gestión del Rendimiento de la Aplicación con prestaciones avanzadas para la monitorización de plataformas, servidores y sistemas informáticos. Ofrece al administrador potentes herramientas de diagnóstico que ayudan a identificar y diagnosticar los problemas antes de que impacten en los usuarios (ManageEngine, 2017).

Cuenta con una sola plataforma integrada para monitorizar todo su ecosistema de aplicaciones y componentes subyacentes de infraestructura tales como servidores web, servicios web, sistemas virtuales entre otros (ManageEngine, 2017). Obtiene notificaciones instantáneas en caso de problemas. Proporciona una disponibilidad en profundidad y la supervisión del rendimiento de los servidores Apache, además del tiempo de respuesta y otras métricas de rendimiento a intervalos regulares de tiempo.

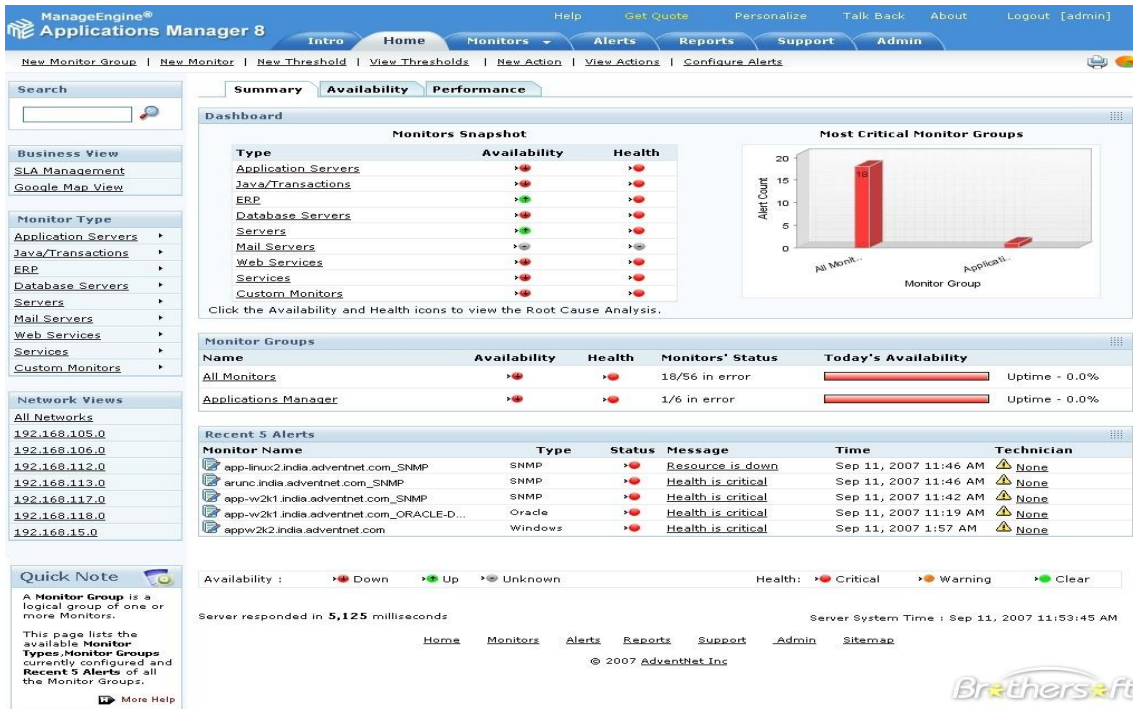


Figura 1 Interfaz de Applications Manager

Webmin

Webmin, véase en la figura 2, es una aplicación web desarrollada por Jamie Cameron y la Comunidad Webmin para la administración de sistemas operativos con distribuciones GNU/Linux de forma local y remota. Este sistema permite la configuración de la mayoría de los servicios mediante módulos entre los cuales se encuentra: Apache WebServer, BIND DNS Server, CVS Server, DHCP Server, FTP, Proxy y Jabber. Incluye un servidor web que funciona con SSL (protocolo HTTPS). Está construido a partir de módulos, los cuales tienen una interfaz a los archivos de configuración, lo que permite añadir nuevas funcionalidades (Webmin, 2016).

Se emplea un módulo para la administración de servicios web Apache, permite configurar casi todas las características de Apache y determinar la versión que está instalada en el sistema. Ajusta su interfaz de usuario para que pueda editar solo aquellas directivas que el servidor web entienda y las agrupa en categorías como direcciones, redes, procesos y límites. Su página principal muestra opciones como iniciar y detener el servicio, restringir el control de acceso al módulo, es decir, otorgar a los clientes los derechos de editar las configuraciones para sus propios servidores, pero no las de los demás. Además, permite editar los archivos en el sistema que se muestran cuando un cliente se conecta al servidor web. Posibilita alojar varios sitios web y crear un host virtual Apache para cada uno (Webmin, 2016).

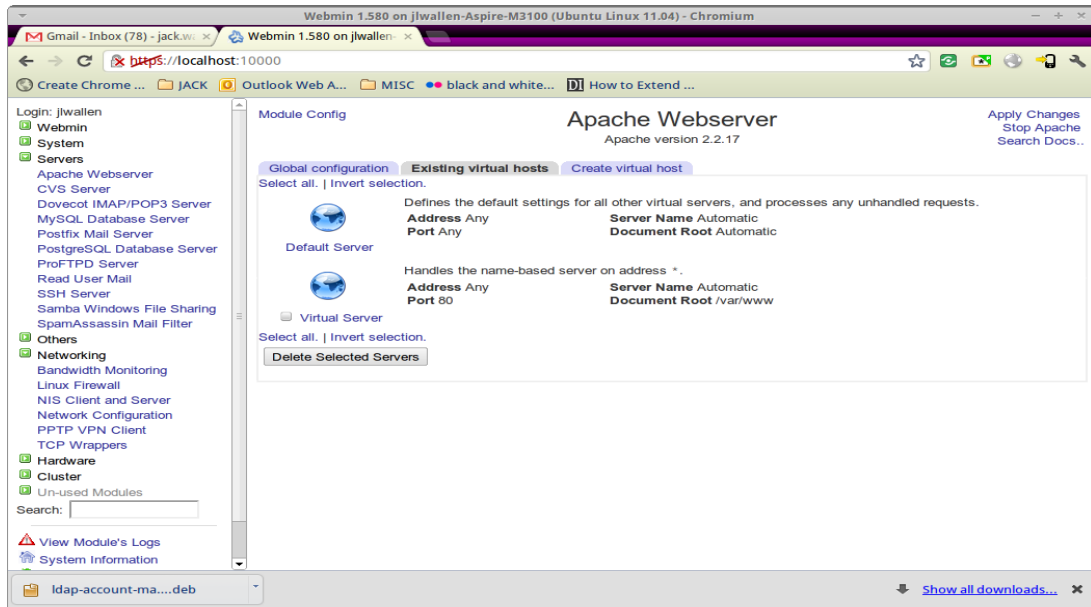


Figura 2 Interfaz del Webmin

Apache GUI

Apache GUI, como se muestra en la figura 3, es una herramienta web gratuita y de código abierto que está diseñada para ayudar a administrar el servidor web Apache. Permite buscar y validar configuraciones. Posibilita administrar los hosts virtuales, instalar y eliminar módulos de Apache, ver las estadísticas de tiempo de ejecución del servidor web, registrar y graficar transacciones detalladas. Posibilita abrir varias pestañas para editar múltiples archivos de configuración al mismo tiempo. La versión más actual de Apache GUI es compatible con los sistemas operativos Solaris 10, Ubuntu, Debian, Fedora, Cent OS, RHEL, Open SUSE, MAC OSX y Windows (ApacheGUI, 2015).

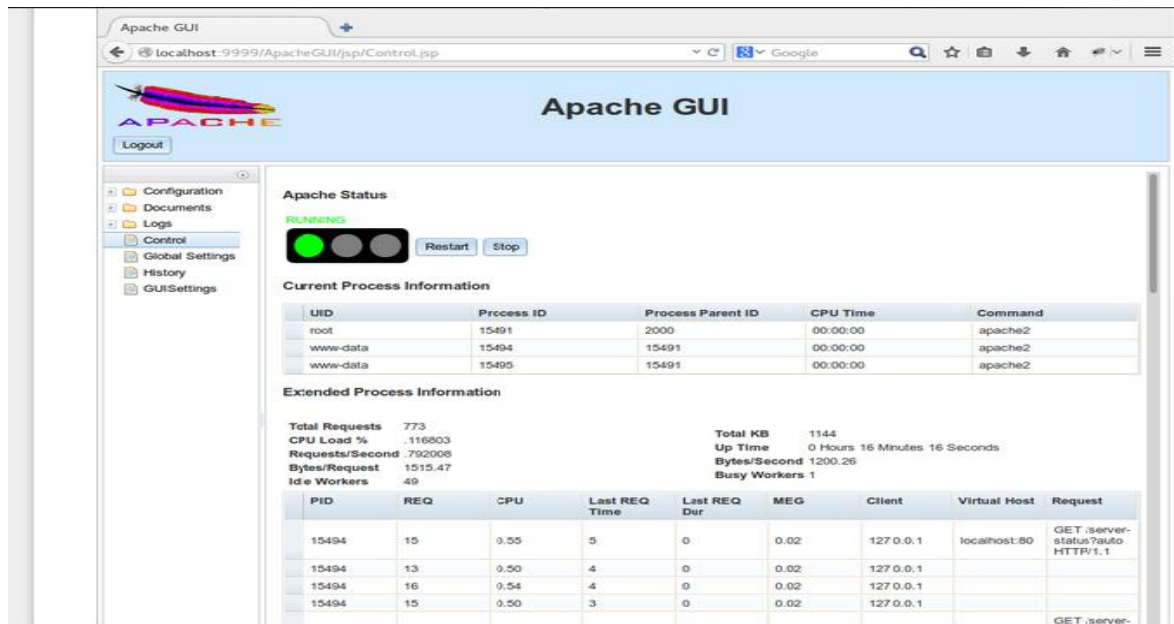


Figura 3 Interfaz de Apache GUI

cPanel

cPanel (control Panel) es una aplicación de administración creada por John Nick Koston en el año 1996. Está basada en la web que se utiliza para administrar sitios de manera fácil, sencilla y con una interfaz muy intuitiva al mismo que se puede ingresar con un usuario y contraseña (Ambuludí, 2014). Este software cuenta con un diseño en tres capas que entrega distintos atributos a administradores, revendedores de espacio y usuarios finales. Estos atributos permiten controlar diversos aspectos de los servidores y servicios entregados a los clientes. El software no es gratuito y se distribuye solamente a los proveedores de alojamiento web.

Con cPanel, como se muestra en la figura 4, es posible manejar todas las opciones del servidor desde un solo lugar. Posibilita el acceso rápido y gestión de los archivos y permite a los usuarios redimensionar, convertir y visualizar imágenes. Muestra un listado completo de visitantes en los archivos de registro de Apache. Permite visualizar un conjunto completo de los errores más recientes en el sistema, personaliza las páginas índice por defecto de Apache y controla los tipos de archivos y extensiones. La última versión de cPanel soporta la instalación en varios sistemas operativos como: CentOS, Red Hat Enterprise Linux (RHEL), y CloudLinux.

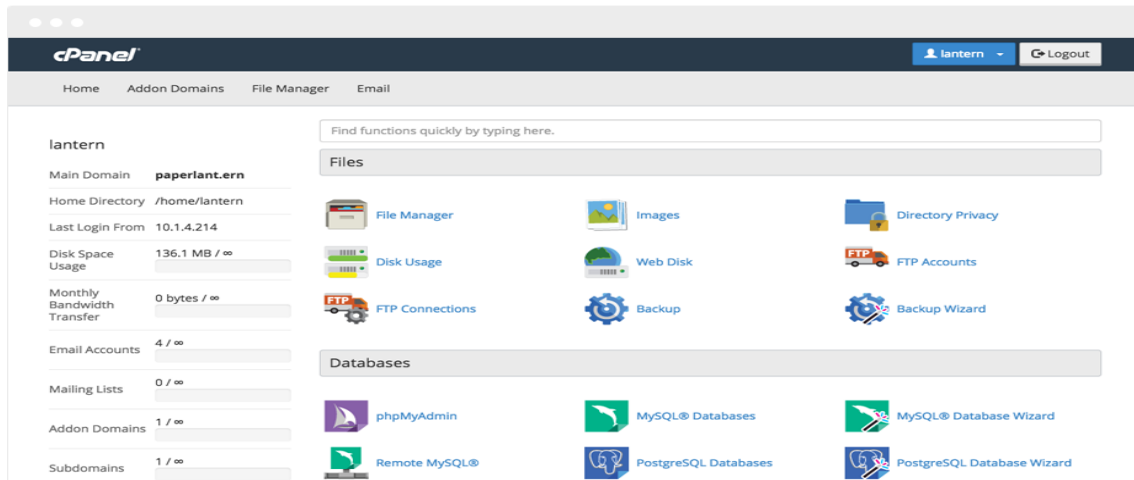


Figura 4 Interfaz de cPanel

Nubity

Nubity, como se muestra en la figura 5, es un software creado por la compañía Nubity en la ciudad de México que integra servidores físicos y virtuales en una sola plataforma de administración, gestión y monitorización. Da soporte a distribuciones de Linux (Debian, Ubuntu, CentOS, Red Hat Enterprise Linux) y Windows Server. Permite monitorear servidores en tiempo real, mostrar el comportamiento de los mismos en gráficos y generar alertas en caso de que ocurran errores. Posibilita saber cuáles son las métricas del servicio web apache, es decir, si se está ejecutando correctamente, si llegó a la cantidad máxima de procesos hijos que tiene permitido abrir y si está recibiendo conexiones de manera correcta (Nubity, 2014).



Figura 5 Interfaz de la Plataforma Nubity

1.2.2 Universidad de las Ciencias Informáticas

HMAST

La Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST) permite la administración de servicios telemáticos con la distribución cubana GNU/Linux Nova, siendo esta la que se emplea en los procesos de migración. HMAST administra sus servicios de forma local y remota. No introduce elementos (procesos, programas, ficheros, base de datos) ajenos o innecesarios para el funcionamiento del servicio telemático en los servidores de manera que se optimice el uso de los recursos en el servidor.

HMAST, véase en la figura 6, puede ser utilizada tanto por usuarios avanzados como por aquellos que poseen conocimientos básicos asociados a los servicios telemáticos en GNU/Linux. Toda la interfaz está en idioma español y las variables de configuración tienen nombres intuitivos que facilitan la inferencia de su funcionalidad. Además, incluye una ayuda que indica los valores admisibles, así como una recomendación de cuál es el valor más adecuado en cada caso. Al introducir errores en el sistema no se guardan los cambios en el servidor y se le notifica al usuario para que sean corregidos, al igual ocurre cuando se ha modificado un archivo.

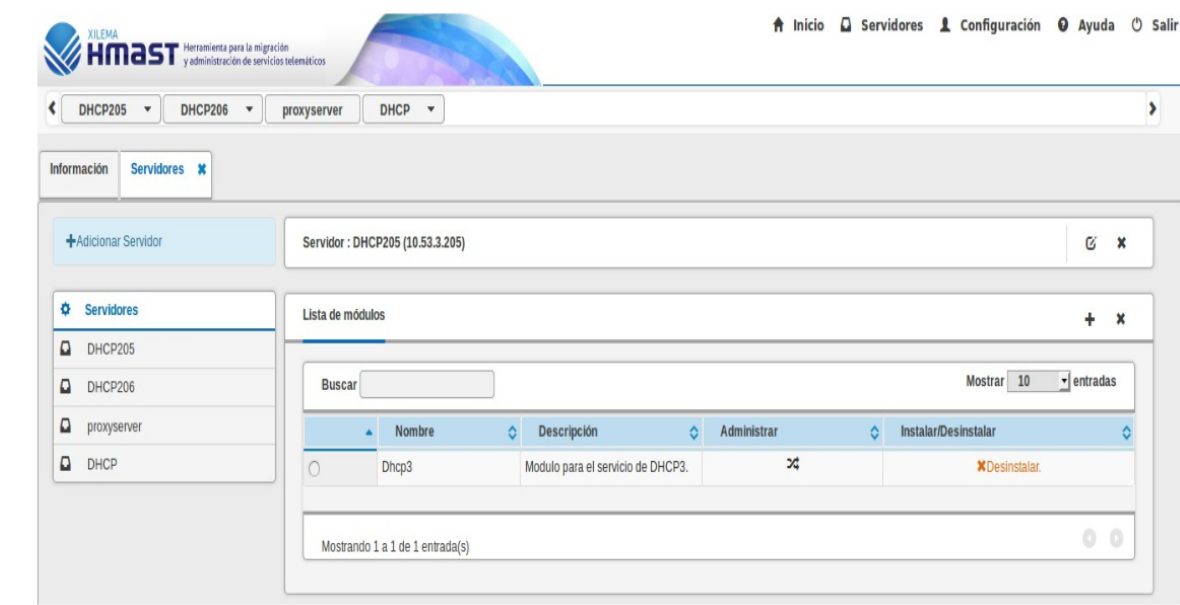


Figura 6 Interfaz de HMAST

Luego de realizar un estudio de las soluciones existentes para la administración, gestión y monitoreo de servicios web apache, se determina no utilizar ninguna de ellas para apoyar la investigación. No cubren

las necesidades específicas identificadas en el Departamento de Componentes del Centro de TLM. En el caso de *Applications Manager*, permite visualizar el rendimiento de los servidores mediante gráficas, debido a que posee buena interfaz gráfica. Sin embargo, presenta inconvenientes, puesto que solo realiza la monitorización de los servicios web Apache. Además de ser un software privativo.

cPanel es otra de las aplicaciones que permite llevar a cabo la administración de servidores web, pero se encuentra bastante limitada, pues no es un software gratuito y no realiza el monitoreo de los servicios web Apache. Nubity es otro de los sistemas para la administración y monitoreo de servicios web. Por otra parte, no puede ser reutilizado, debido a que es un software privativo.

En cuanto a los softwares libres, Webmin es una de las soluciones de código abierto útiles para administrar los servicios web apache y realizar configuraciones necesarias en el servidor. No obstante, presenta algunas desventajas como, por ejemplo, la no administración de varios servidores de manera simultánea, sería necesario la instalación de Webmin en el sistema operativo de cada uno de estos servidores. Otro de los problemas que presenta es que se sobrescribe el fichero de configuración al realizarse algún cambio en la interfaz gráfica.

HMAST es otra de las herramientas libres que permite realizar la administración de los servicios web apache. Por otro lado, tiene sus limitaciones, pues no permite restringir el acceso de usuarios remotos a los archivos de configuración de Apache. No cuenta con una funcionalidad que permita mostrar información acerca de los Logs de acceso y el registro de errores en el servicio Apache. Además, no realiza la monitorización de dichos servicios. Al igual que HMAST, Apache GUI permite administrar servicios web, pero no realiza la monitorización de los mismos que se encuentran en los servidores. Por estas razones no se elige ninguna de estas aplicaciones como solución al problema planteado en esta investigación. Se decide implementar un sistema informático que contará con las funcionalidades necesarias y capaz de satisfacer las necesidades específicas del área. Para el desarrollo de dicho sistema se tomaron en cuenta algunas ideas de la herramienta HMAST, por lo que se decide utilizar de la misma: algunas de sus funcionalidades básicas para la administración de Apache y aspectos regulares en la estructura de los formularios.

1.3 Metodología de Desarrollo de Software

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Se han desarrollado a lo largo de los años una gran variedad de metodologías, cada una de ellas con sus fortalezas y debilidades. Dentro de estas metodologías existen dos grandes grupos, las metodologías tradicionales y las ágiles (Romero, 2012).

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación. Por otra parte, las metodologías ágiles conllevan a que el cliente y los desarrolladores trabajen en conjunto con una cercana comunicación. Posibilita la entrega continua y en plazos cortos de software funcional. Las metodologías ágiles ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Una determinada metodología no es necesariamente aplicable a todo tipo de proyectos, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor (Hernández, 2014).

1.3.1 Programación Extrema (XP)

La Programación Extrema (*eXtreme Programming*, XP, por sus siglas en inglés) es uno de los ejemplos más exitosos de metodología ágil. XP nace como una nueva disciplina de desarrollo de software creada por Kent Beck. Metodología de desarrollo ligera basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software.

Promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se fundamenta en la retroalimentación continua entre el cliente y el equipo de desarrollo. Comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define para proyectos con requisitos imprecisos, muy cambiantes, y donde existe un alto riesgo técnico (Montoya Suarez, 2017).

Para el desarrollo del presente trabajo de diploma se decide utilizar la metodología de desarrollo de software XP, debido a que:

- El cliente forma parte del equipo de desarrollo.
- Consta de una alta gama de bibliografía.
- Sus artefactos presentan una descripción del comportamiento del sistema fácil de entender.
- Los requisitos tienden a cambiar frecuentemente, según vaya avanzando el trabajo.
- Es el más destacado de los procesos ágiles de desarrollo de software.

1.4 Herramientas y lenguajes informáticos

1.4.1 Lenguaje de programación del lado del servidor

Python 3.6.0

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90. Es un lenguaje potente y fácil de aprender. Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos, es construido sobre objetos que combinan datos y funcionalidades. La sintaxis de Python, su gestión de tipos dinámica y su naturaleza interpretada hacen de éste un lenguaje ideal para el desarrollo rápido de aplicaciones, en muchas áreas y en la mayoría de las plataformas (Rossum, 2017). Se decide utilizar Python 3.6.0 porque que facilita el trabajo con expresiones, clases definidas por el usuario, instrucciones y tipos de datos simples. Permite separar el sistema en módulos, propiciando una mayor organización del trabajo realizado. Promueve un estilo de codificación fácil de leer y visualmente agradable.

1.4.2 Lenguajes de programación del lado del cliente

HTML 5

HTML (*HyperText Markup Language*) es un lenguaje sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada, con enlaces que conducen a otros documentos o fuentes de información relacionadas. Es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes y listas (Gauchat, 2012). Se decide utilizar HTML5 debido a que incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo, por lo que las secciones más importantes son diferenciadas y la estructura principal ya no depende de los elementos como tablas y etiquetas div.

CSS 3

CSS (*Cascade Style Sheets*), también llamado hojas de estilo en cascada, es un lenguaje de marcado que se emplea para dar formato a un sitio web. Es decir, funciona en conjunto con los archivos HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes entre otros. Permite separar la estructura de la presentación del documento (Gauchat, 2012). Se decide utilizar CSS 3 para la aplicación de estilos que se muestran en el sistema a desarrollar porque permite organizar el documento principal. Permite incrementar la velocidad de carga y aprovechar las nuevas características de HTML5, propiciando un entorno del sistema más atractivo y amigable para los usuarios.

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar *ECMAScript* (Asociación Europea de Fabricantes de Computadoras). Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Javascript es soportado por la mayoría de los

navegadores como *Internet Explorer, Netscape, Opera y Mozilla Firefox*. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario (Castillo, 2017). Se decide utilizar JavaScript para el desarrollo de la solución propuesta porque permite controlar las ventanas del navegador y el contenido que muestran. Permite comprobar los datos que el usuario introduce en un formulario antes de enviarlos.

1.4.3 Framework de desarrollo

La palabra framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular. Sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida. Presenta artefactos o módulos de software concretos. Contiene una base en la cual otro proyecto de software puede ser organizado y desarrollado. 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 (Rodríguez, 2011).

Django 1.10.8

Django es un marco de desarrollo en Python de código abierto, legible que permite la creación rápida de páginas y aplicaciones web. Utiliza una variación de la arquitectura MVC (Modelo-Vista-Controlador) llamada MTV (Modelo-Plantilla-Vista). Impulsa el desarrollo de código limpio al promover buenas prácticas de desarrollo web. Contiene su propio ORM (mapeo-objeto-relacional), una capa intermedia entre el código y la base de datos que separa la integración entre ambos por completo. Ofrece una serie de soluciones a problemas comunes, basándose en el uso de patrones de diseño (Django, 2016). Por lo anteriormente expuesto se decide utilizar Django como framework de desarrollo para la implementación de la solución propuesta.

jQuery 1.9.1

jQuery es una librería JavaScript de código abierto, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación mucho más fácil y rápida del lado del cliente. Integra funcionalidades para trabajar con AJAX (*Asynchronous JavaScript And XML*, por sus siglas en inglés) o JavaScript asíncrono y XML. Provee de un mecanismo para la captura de eventos (Castillo, 2017).

Se decide utilizar JQuery en su versión 1.9.1 para el desarrollo del sistema propuesto porque facilita la selección de elementos HTML. Permite cambiar la información de una página web sin necesidad de recargarla. Provee soporte para viejos navegadores y puede ser utilizado como una forma simple de reemplazar funciones de HTML5.

Bootstrap 3.3.1

Bootstrap, es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Es una herramienta ágil para construir sitios web (Bootstrap, 2015). Se escoge Bootstrap en su versión 3.3.1 para el desarrollo de la solución propuesta debido a que se integra con las librerías JavaScript. Permite la realización de formularios, menús, botones, la utilización de íconos y otros componentes para el desarrollo de las aplicaciones web.

1.5 Visual Paradigm 8.0

Es una herramienta CASE (*Computer Aided Software Engineering*) o Ingeniería de Software Asistida por Computadora. Visual Paradigm 8.0 puede ayudar en todos los aspectos del ciclo de vida de desarrollo del software: análisis y diseño orientado a objetos. Posibilita el desarrollo de la ingeniería inversa y directa. Se encuentra disponible en múltiples plataformas como Linux y Microsoft Windows. Permite realizar diagramas para el diseño enfocado a los procesos del negocio y exportar e importar componentes como imágenes XML (Visual Paradigm, 2018).

1.6 Notación para el Modelado de Procesos del Negocio (BPMN)

Business Process Model and Notation (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara y completa. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD, por sus siglas en inglés) (bizagi, 2014). Permite expresar los procesos en un único diagrama de proceso de negocio. Por lo anteriormente expuesto se decide escoger Notación para el Modelado de Procesos de Negocio BPMN.

1.7 Entorno de Desarrollo Integrado

Un IDE (Entorno de Desarrollo Integrado) es una aplicación que ofrece amplias facilidades para un equipo de programadores durante el desarrollo de software. Un IDE normalmente consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador. Varios IDEs modernos se integran con características de codificación inteligentes (IDE, 2015). Uno de los objetivos del IDE es reducir la configuración necesaria para juntar múltiples utilidades de desarrollo, y ofrecer el mismo conjunto de capacidades como una unidad coherente.

PyCharm 2017.3.3

PyCharm es desarrollado por la empresa JetBrains. Es un entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica y soporte para el desarrollo web con Django. Soporta varias bases de datos, entornos virtuales e intérpretes de Python 2.x, 3.x (JetBRAINS, 2017). Se decide utilizar PyCharm 2017.3.3 como entorno de desarrollo integrado porque permite la integración con frameworks web como Django. Posibilita el autocompletado de código, el resaltado de sintaxis y de errores.

1.8 Sistema Gestor de Base de Datos

Es un programa orientado a la gestión y diseño de bases de datos, permitiendo su creación, modificación, atributos e interfaz, interactuar y explotar sus contenidos. El objetivo principal de un SGBD (Sistema Gestor de Base de Datos) es proporcionar un entorno eficaz a la hora de almacenar y recuperar la información de la base de datos. (Cobo, 2007). Permite a los usuarios una visión abstracta de la información, es decir, el sistema ahorra al usuario la necesidad de conocer al detalle cómo se almacenan los datos.

SQLite 3

SQLite es una biblioteca en proceso que implementa un motor de base de datos SQL transaccional independiente, sin servidor y de configuración cero. SQLite lee y escribe directamente en archivos de disco ordinarios. Una base de datos SQL completa con múltiples tablas, índices, disparadores y vistas. Está contenida en un solo archivo de disco. El formato de archivo de la base de datos es multiplataforma: puede copiar libremente una base de datos entre sistemas de 32 bits y de 64 bits (SQLite, 2018). Se escoge SQLite 3 como gestor de bases de datos debido a que viene integrado a Python, sus bases de datos pueden ser portadas sin ninguna configuración o administración. Cuenta con diferentes interfaces API (Interfaces de Programación de Aplicaciones), las cuales permiten trabajar con Python.

Conclusiones del capítulo

En este capítulo se realizó un estudio del estado del arte de sistemas homólogos relacionados con la gestión, administración y monitorización de los servicios web Apache. Como resultado del estudio se obtuvo que ninguno de los sistemas anteriormente expuestos cumple con las necesidades que requiere el problema planteado en esta investigación. Es por ello que se decide implementar un sistema informático para la gestión, administración y monitorización de servicios web apache. Para guiar el proceso de desarrollo del software se seleccionó como metodología XP, propiciando una mayor organización en el diseño y generando los elementos esenciales para el desarrollo del sistema. Además, se definieron las herramientas y lenguajes informáticos a emplear, teniendo en cuenta sus características para contribuir al desarrollo de la propuesta de solución.

Capítulo 2: Análisis y diseño del sistema

Introducción

En el presente capítulo se describe la solución propuesta del sistema a implementar, con el objetivo de satisfacer las necesidades del Departamento de Componentes del Centro de TLM. Se describe el flujo actual del proceso que se lleva a cabo en dicho departamento para la gestión, administración y monitorización de los servicios web Apache. Se abordan los principales artefactos generados por la metodología XP (Programación Extrema) en las fases de exploración, planificación y diseño.

2.1 Proceso de administración de servicios web Apache

El proceso para la gestión y administración del servicio web Apache dentro del Departamento de Componentes del Centro de TLM comienza con la instalación de este servicio. Luego, el administrador realiza la configuración de Apache a través del archivo principal de configuración apache2.conf. Se gestionan los hosts virtuales para acceder a los sitios web alojados en el servicio Apache y se maneja información referente a los registros de accesos y de errores que puedan ocurrir durante la operación del mismo. Posteriormente se reinicia el servicio mediante la consola. A continuación, en la figura 7 se representa el proceso que se lleva a cabo en el departamento para la gestión y administración de los servicios web Apache.

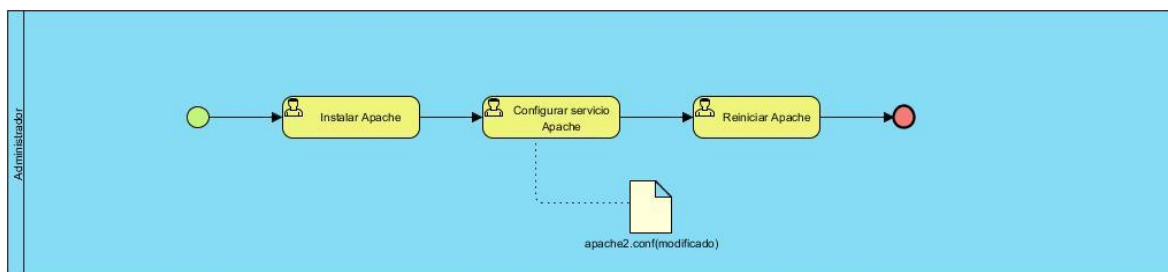


Figura 7 Proceso del negocio actual

Tabla 1 Ficha de proceso

Ficha de proceso	
Proceso	Proceso del negocio actual para la gestión y administración del servicio web Apache.
Descripción	Proceso que se realiza actualmente en el Departamento de Componentes del Centro de TLM para la gestión y

	administración del servicio web Apache.
Entradas	Usuario y contraseña del administrador para acceder al sistema.
Salidas	Configuraciones del servicio web Apache.
Reglas del negocio	Se necesita tener instalado el servicio web Apache para posteriormente realizar la configuración del mismo.
Actividades principales	
Actividad 1	Instalar Apache: El administrador debe instalar el servicio web Apache.
Actividad 2	Configurar el servicio Apache: El administrador puede realizar las configuraciones a través del archivo apache2.conf.
Actividad 3	Reiniciar Apache: El Administrador puede reiniciar el servicio Apache.

2.2 Características de la propuesta de solución

La presente investigación propone la implementación de un módulo para la gestión, administración y monitorización de los servicios web Apache para el Departamento de Componentes del Centro de TLM. Solo los usuarios definidos con el rol de administrador podrán acceder al sistema. El módulo realizará la administración remota de Apache en diferentes PC servidoras y permitirá establecer las conexiones seguras a través del protocolo de seguridad SSH (*Secure Shell*). Permitirá además la gestión de hosts virtuales y mostrar de forma estructurada el almacenamiento lógico de los mismos para un mejor entendimiento por parte del usuario, así como la transmisión segura de información a través del protocolo SSL (*Secure Sockets Layers*).

Cuenta con la capacidad de analizar el fichero de accesos y el registro de errores, lo que posibilitará mostrar información detallada sobre las peticiones que el servicio web recibe. También permitirá modificar los puertos de escucha del servicio Apache. Entre sus funciones se encuentran instalar Apache, y una vez instalado podrá manejar los estados del mismo, así como desinstalarlo. El sistema mostrará gráficas sobre el estado de los servicios web Apache, permitiendo conocer cuáles y que cantidad de recursos son utilizados en el mismo. Otra de sus funcionalidades es la de restringir acceso a Apache, donde el

administrador podrá otorgar permisos de edición a los archivos de configuración de Apache a los usuarios remotos. Permitirá la gestión de máquinas servidoras y contará con un módulo “Usuario”, en el cual podrán gestionarse los usuarios con roles de administrador en el sistema. El diseño del mismo estará basado en la Estrategia Marcaria que establece la UCI.

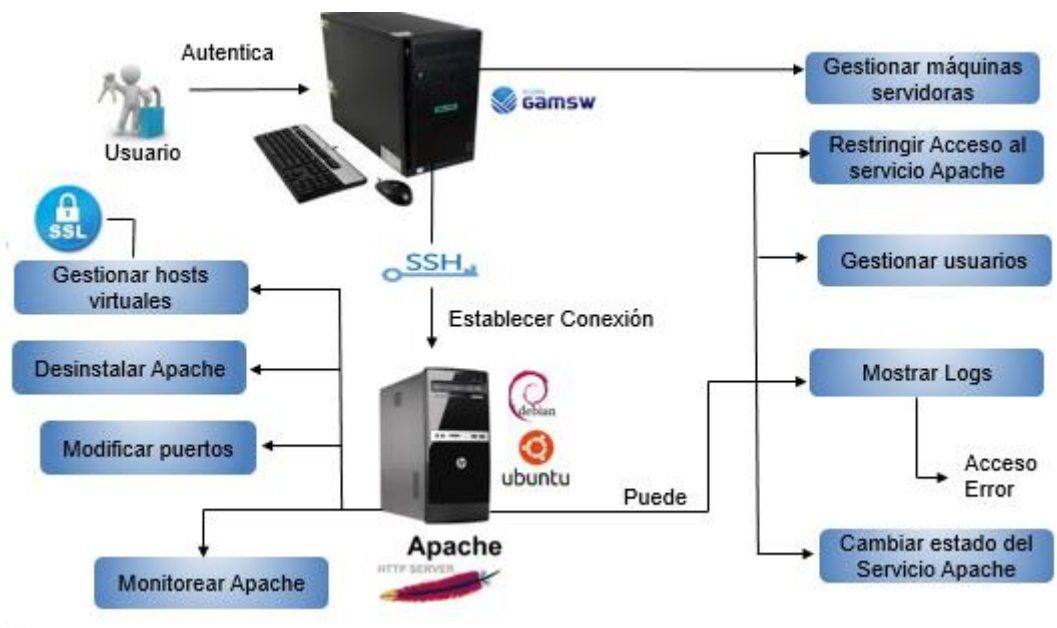


Figura 8 Propuesta de solución del sistema

2.3 Roles y funcionalidades

Luego de haber realizado entrevistas a los especialistas del Departamento de Componentes y teniendo en cuenta los resultados del estudio de sistemas homólogos, se determinan un conjunto de funcionalidades que especifican lo que debe proporcionar el sistema.

Rol: Administrador del sistema

Es la persona que tiene la responsabilidad de gestionar, administrar, monitorizar, y asegurar el correcto funcionamiento del servicio web Apache, o algún aspecto de éste.

Funcionalidades del sistema

F1 Autenticar usuario

F2 Gestionar máquinas servidoras

F3 Establecer conexión por SSH a un servidor remoto

F4 Instalar Apache

F5 Cambiar estado del servicio web Apache

F6 Desinstalar Apache

F7 Restringir acceso al servicio Apache

F8 Gestionar Hosts virtuales

F9 Modificar puertos para las conexiones en Apache

F10 Activar y desactivar SSL

F11 Mostrar Logs

F12 Monitorear servicio Apache

F 13 Gestionar usuario

2.4 Fase de Exploración

El ciclo de vida de XP comienza con la fase de exploración en la cual los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (Letelier, 2006).

2.4.1 Historias de Usuario

Las Historias de usuario sustituyen a los documentos de especificación funcional, y a los casos de uso. Estas historias son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios (Joskowicz, 2008).

En las Historias de Usuario (HU) se considera:

- La prioridad en el negocio:
 - Alta: Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio

- Media: Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- Baja: Cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.
- El riesgo en desarrollo:
 - Alto: Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
 - Medio: Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
 - Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.
- Un Punto de Estimación equivale a una semana de programación, una semana de programación corresponde a 40 horas en desarrollo, 8 horas durante 5 días de la semana del calendario normal. A continuación, se muestra la descripción de algunas de las historias de usuario realizadas, las restantes pueden consultarse en el **Anexo 2**.

Tabla 2 Historia de usuario "Autenticar usuario"

Historia de Usuario	
Número: 1	Nombre: Autenticar usuario
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.2	Iteración asignada: 1
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al usuario introducir el nombre de usuario y contraseña para acceder al sistema.	
Observaciones: El usuario debe estar registrado como activo para acceder al sistema.	
Prototipos de interfaz:	



Tabla 3 Historia de usuario "Gestionar máquinas servidoras"

Historia de Usuario	
Número: 2	Nombre: Gestionar máquinas servidoras
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al usuario adicionar, editar, listar y eliminar máquinas servidoras en el sistema.	
Observaciones: El usuario debe estar autenticado en el sistema.	
Prototipos de interfaz:	

Tabla 4 Historia de usuario "Establecer conexión por SSH a un servidor remoto"

Historia de Usuario

Número: 3	Nombre: Establecer conexión por SSH a un servidor remoto
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.2	Iteración asignada: 1
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al administrador conectarse a un servidor remoto registrado en el sistema.	
Observaciones: El usuario debe estar autenticado en el sistema. El servidor remoto debe tener instalado un servidor SSH.	
Prototipos de interfaz: No tiene interfaz visual.	

Tabla 5 Historia de usuario “Instalar servicio Apache”

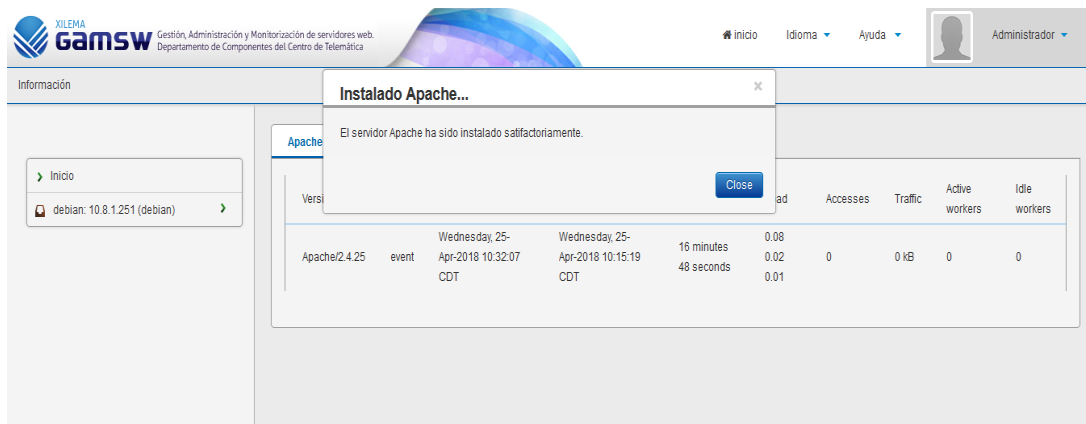
Historia de Usuario																									
Número: 4	Nombre: Instalar servicio Apache																								
Usuario: Administrador																									
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto																								
Puntos estimados: 0.2	Iteración asignada: 1																								
Programador responsable: Emilio Gort Caballero																									
Descripción: Permite al administrador instalar el servicio Apache.																									
Observaciones: El usuario debe estar autenticado en el sistema. Deben estar configurados los repositorios en el servidor remoto para poder instalar el servicio Apache.																									
Prototipos de interfaz:																									
 <p>The screenshot shows a web management interface for 'GAMSW' (Gestión, Administración y Monitorización de servidores web). A modal window titled 'Instalado Apache...' is open, displaying the message 'El servidor Apache ha sido instalado satisfactoriamente.' Below the modal, a table shows system logs for the Apache service.</p> <table border="1"> <thead> <tr> <th>Vers</th> <th>ad</th> <th>Accesses</th> <th>Traffic</th> <th>Active workers</th> <th>Idle workers</th> </tr> </thead> <tbody> <tr> <td>Apache/2.4.25</td> <td>event</td> <td>0</td> <td>0 kB</td> <td>0</td> <td>0</td> </tr> <tr> <td></td> <td>Wednesday, 25-Apr-2018 10:32:07 CDT</td> <td>0.02</td> <td></td> <td>16 minutes</td> <td>0.08</td> </tr> <tr> <td></td> <td>Wednesday, 25-Apr-2018 10:15:19 CDT</td> <td>0.01</td> <td></td> <td>48 seconds</td> <td>0.01</td> </tr> </tbody> </table>		Vers	ad	Accesses	Traffic	Active workers	Idle workers	Apache/2.4.25	event	0	0 kB	0	0		Wednesday, 25-Apr-2018 10:32:07 CDT	0.02		16 minutes	0.08		Wednesday, 25-Apr-2018 10:15:19 CDT	0.01		48 seconds	0.01
Vers	ad	Accesses	Traffic	Active workers	Idle workers																				
Apache/2.4.25	event	0	0 kB	0	0																				
	Wednesday, 25-Apr-2018 10:32:07 CDT	0.02		16 minutes	0.08																				
	Wednesday, 25-Apr-2018 10:15:19 CDT	0.01		48 seconds	0.01																				

Tabla 6 Historia de usuario “Cambiar estado del servicio Apache”

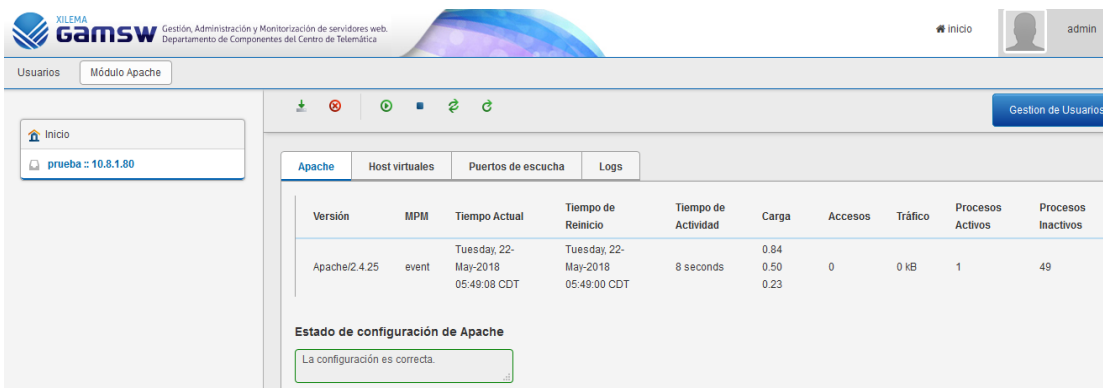
Historia de Usuario	
Número: 5	Nombre: Cambiar estado del servicio Apache
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al administrador realizar acciones como iniciar, detener, reiniciar y recargar el servicio Apache.	
Observaciones: El servicio Apache debe estar instalado.	
Prototipos de interfaz:	
	

tabla 7 Historia de usuario “Desinstalar servicio Apache”

Historia de Usuario	
Número: 6	Nombre: Desinstalar servicio Apache
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al administrador desinstalar el servicio Apache.	
Observaciones: El servicio Apache debe estar instalado. El usuario debe estar autenticado en el sistema.	
Prototipos de interfaz:	

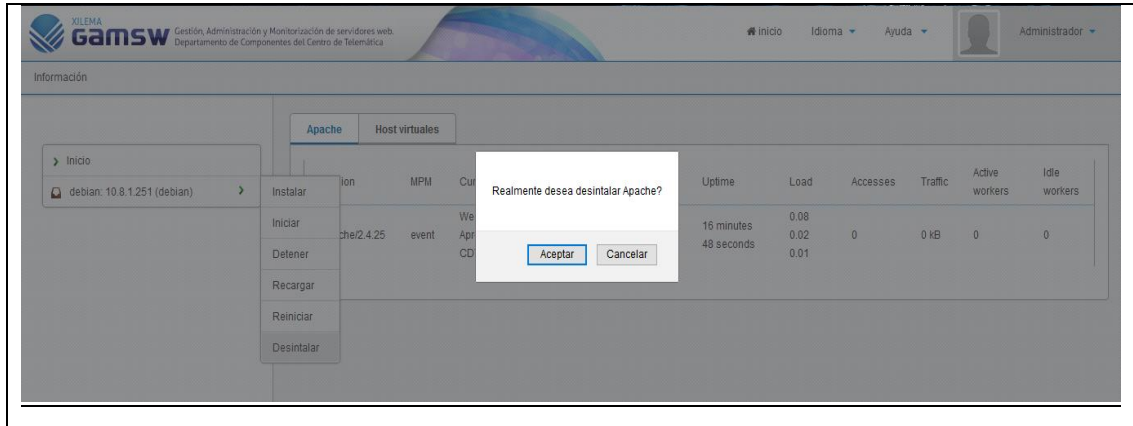


tabla 8 Historia de usuario “Gestionar Hosts Virtuales”

Historia de Usuario	
Número: 8	Nombre: Gestionar Hosts Virtuales
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al administrador adicionar, eliminar, modificar, habilitar y deshabilitar hosts virtuales en el servidor Apache.	
Observaciones: El servicio Apache debe estar instalado. El usuario debe estar autenticado en el sistema.	
Prototipos de interfaz:	

tabla 9 Historia de usuario “Mostrar Logs”

Historia de Usuario

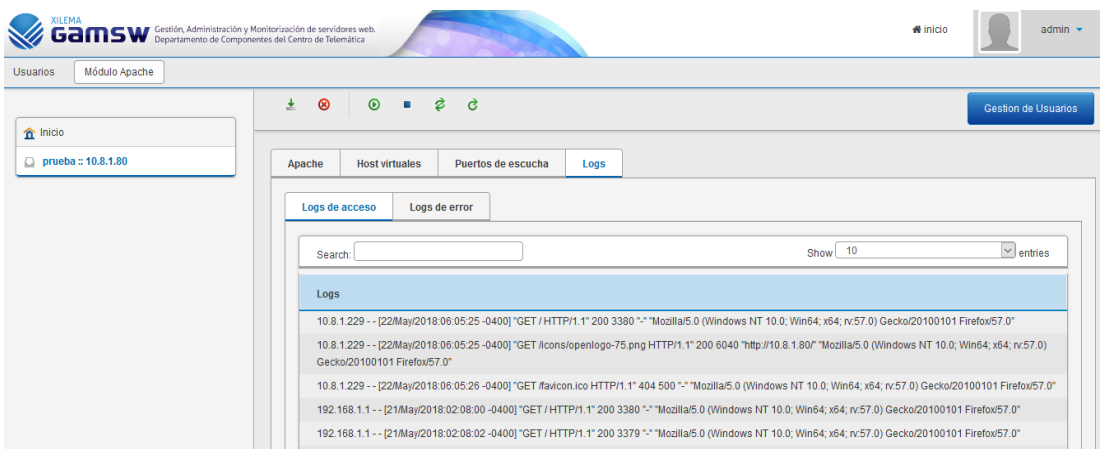
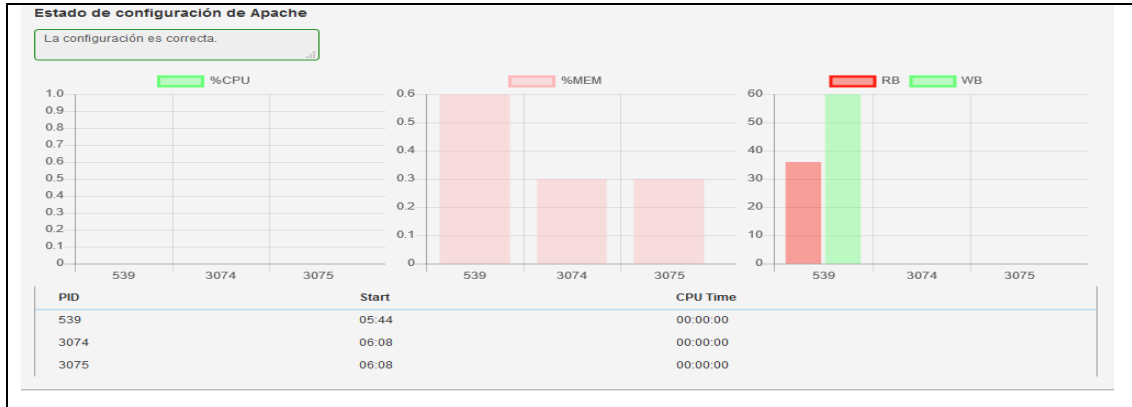
Número: 11	Nombre: Mostrar Logs
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Emilio Gort Caballero	
Descripción: Permite mostrar información al administrador sobre los Logs de accesos y el registro de errores.	
Observaciones: El servicio Apache debe estar instalado. Tienen que estar creados los Logs correspondientes para cada virtual host.	
Prototipos de interfaz:	
	

tabla 10 Historia de usuario “Monitorear servicio Apache”

Historia de Usuario	
Número: 12	Nombre: Monitorear servicio Apache
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.2	Iteración asignada: 3
Programador responsable: Emilio Gort Caballero	
Descripción: Permite al administrador mostrar el uso de los recursos del servidor utilizados por el servicio Apache, como CPU, memoria RAM y la cantidad de bytes escritos y leídos por los procesos de Apache.	
Observaciones: El servicio Apache debe estar instalado. El usuario debe estar autenticado en el sistema.	
Prototipos de interfaz:	



2.5 Fase de planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. El resultado de esta fase es un Plan de Entrega. Una vez acordado este plan, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas historias de usuarios (Joskowicz, 2008).

2.5.1 Estimación del esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la velocidad de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración (Letelier, 2006). A continuación, se muestran los resultados de la estimación de esfuerzo por cada una de las historias de usuario:

Tabla 11 Estimación de esfuerzo

Iteración	Historias de Usuario		Puntos estimados (semanas)
1	1	Autenticar usuario	0.2
	2	Gestionar máquinas servidoras	2
	3	Establecer conexión por SSH a un	0.2

		servidor remoto	
	4	Instalar servicio Apache	0.2
	5	Cambiar estado del servicio Apache	0.4
2	6	Desinstalar Apache	0.4
	7	Restringir acceso del servicio Apache	0.6
	8	Gestionar Hosts Virtuales	1
	9	Modificar puertos para las conexiones en Apache	1
3	10	Activar y desactivar SSL	0.6
	11	Mostrar Logs	0.2
	12	Monitorear servicio Apache	0.2
	13	Gestionar usuario	2
Total			9

Luego de realizar una estimación de esfuerzo de las historias de usuarios se puede concluir que las 13 historias de usuarios asociadas en 3 iteraciones demoran en realizarse 9 semanas.

2.5.2 Plan de iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible debido a que el cliente es quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Iteración 1

En esta iteración se implementan las historias de usuarios con un nivel de prioridad alto establecido por el cliente por su alto valor del negocio, creando con ellas el funcionamiento base del sistema. Luego de concluir la iteración se podrán probar funcionalidades como: Autenticar usuario, gestionar máquinas servidoras, establecer conexión por SSH a un servidor remoto, instalar servicio Apache, y cambiar estado del servicio Apache especificadas en las historias de usuario 1, 2, 3, 4, 5.

Iteración 2

En esta iteración se implementan las historias de usuarios con un nivel de prioridad medio establecido por el cliente por su impacto en el negocio, relacionadas con la segunda etapa del proceso de gestión, administración y monitorización del servicio Apache. Luego de concluir la iteración se podrán probar funcionalidades como: Desinstalar servicio Apache, restringir acceso al servicio Apache, gestionar Hosts Virtuales y modificar puertos para las conexiones en Apache especificadas en las historias de usuario: 6, 7, 8, 9.

Iteración 3

En esta iteración se implementan las historias de usuarios con un nivel de prioridad bajo establecido por el cliente por su impacto en el negocio, creando con ellas un funcionamiento completo del sistema. Luego de concluir la iteración se podrán probar funcionalidades como: Activar y desactivar SSL, analizar Logs, monitorear servicio Apache y gestionar usuario, especificadas por las historias de usuario 10, 11, 12, 13.

Tabla 12 Plan de duración de las iteraciones

Iteración	Historias de Usuario		Duración (semanas)
1	1	Autenticar usuario	3
	2	Gestionar máquinas servidoras	
	3	Establecer conexión por SSH a un servidor remoto	
	4	Instalar servicio Apache	
	5	Cambiar estado del servicio Apache	
2	6	Desinstalar servicio Apache	3
	7	Restringir acceso del servicio Apache	
	8	Gestionar Hosts Virtuales	
	9	Modificar puertos para las conexiones en Apache	
3	10	Activar y desactivar SSL	3
	11	Mostrar Logs	
	12	Monitorear servicio Apache	
	13	Gestionar usuario	
Total			9

2.5.3 Plan de Entrega

El plan de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este plan será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores y gerentes). El cliente ordenará y agrupará según sus prioridades las historias de usuario. El plan de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario (Letelier, 2006). En la siguiente tabla se muestra la fecha en la que se va a efectuar cada una de las entregas de las iteraciones del sistema al cliente.

Tabla 13 Plan de Entrega

Iteración	Fecha de entrega
Iteración 1	30 de marzo de 2018
Iteración 2	20 de abril de 2018
Iteración 3	11 de mayo de 2018

2.6 Fase de diseño

La metodología XP hace especial énfasis en el diseño de una solución que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos (Letelier, 2006).

2.6.1 Patrones de Arquitectura

Son patrones de software los cuales se encargan de definir la estructura de un sistema, estos a su vez se componen de subsistemas con sus responsabilidades. También tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema. Un patrón arquitectónico se enfoca a dar solución a un problema en específico, de un atributo de calidad, y abarca solo parte de la arquitectura (Tinoco, 2013).

Arquitectura Cliente Servidor

Se decide utilizar la arquitectura Cliente Servidor porque:

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red (Vila, 2018).

Entre las características básicas de una arquitectura Cliente/Servidor se encuentran la combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos e impresoras. Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, capacidades del disco y dispositivos de entrada y salida. Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo, pues esperan las peticiones de los clientes (Contreras Ortiz, 2017).

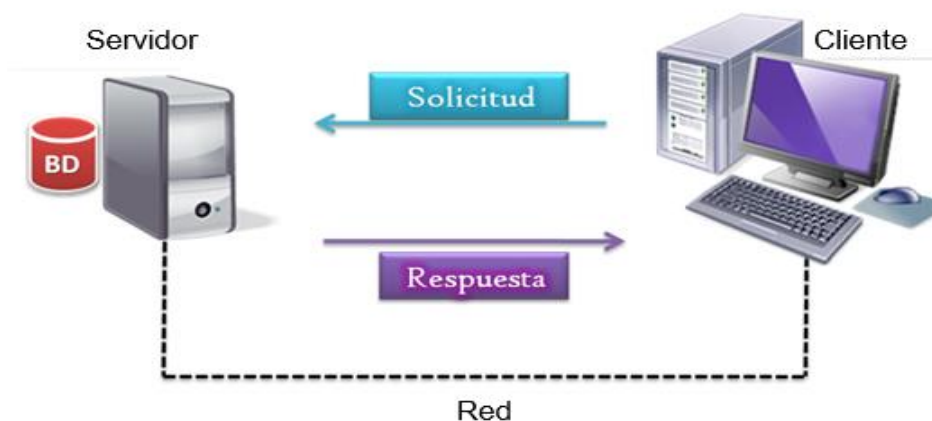


Figura 9 arquitectura Cliente-Servidor

Model Template View (MTV)

Para el desarrollo del sistema se decide utilizar el *Model Template View* (MTV por sus siglas en inglés) porque es un estilo de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

El Modelo

El modelo define los datos almacenados. Se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos (Infante Montero, 2012). En el modelo se representa la clase Servidor.

La Vista

La vista se presenta en forma de funciones en Python, su propósito es determinar qué datos serán visualizados. El ORM (*Object-Relational Mapping*) de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo más importante a entender con respecto a la vista es que no tiene nada que ver con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla (Infante Montero, 2012). En la vista se encuentran las funciones implementadas que se mostrarán con la plantilla, dentro de estas funciones se encuentran: `edit_server`, `virtual_host` y `add_server`.

La Plantilla

La plantilla recibe los datos de la vista y luego los organiza para la presentación al navegador web. Las etiquetas que Django usa para las plantillas permiten que sea flexible para los diseñadores del *frontend*, incluso tiene estructuras de datos como `if`, por si es necesaria una presentación lógica de los datos, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código Python. Esto permite que la lógica del sistema siga permaneciendo en la vista (Infante Montero, 2012). En la plantilla se representan las vistas que mostrarán la interfaz por la cual se administrarán los servicios Apache, como, por ejemplo: `server_admin.html`, `listado_pc.html` y `remote_users.html`.

El uso de los frameworks basados en el patrón de arquitectura MTV, véase en la figura 10 permite tener una separación lógica y física de los componentes de la aplicación. Además de contribuir a una elevada organización en el trabajo y que el proyecto sea bien desarrollado.

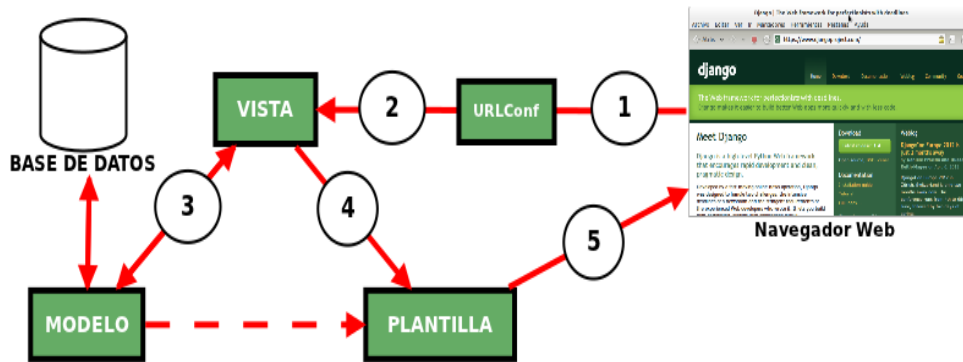


Figura 10 MTV

2.7 Patrones de Diseño

Los patrones de diseño consisten en la descripción de problemas recurrentes en la programación orientada a objetos, dentro de un contexto dado y la definición de una posible solución correcta al mismo. También puede considerarse como un documento que define una estructura de clases que aborda una situación particular (CCM, 2016).

Patrones GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*, por sus siglas en inglés) o Patrones de Software de Asignación de Responsabilidad General son parejas de problema-solución con un nombre que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades a objetos, expresados en forma de patrones (Collazo, 2015). A continuación, se definen los patrones de asignación de responsabilidades que se implementan en el software a desarrollar:

Experto en Información

Guía la idea de a que objetos hay que asignar responsabilidades. La respuesta sería al experto en información, es decir, a la clase que tiene la información necesaria para realizar la responsabilidad (Larman, 2003). En realidad, se trata de un principio de guía básico que se ha de utilizar siempre en el diseño de objetos. Se evidencia en la clase `ApacheServerAdmin`, porque es la encargada de realizar las operaciones referentes al servicio Apache.

Creador

Guía la idea de que objeto sería el responsable de la creación de una instancia de alguna clase, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello (Larman, 2003). Este patrón se evidencia dentro del archivo `remote_users.py` en la

funcionalidad RemoteUsers porque dentro de la misma se crean instancias de la clase ApacheServerAdmin.

Bajo Acoplamiento

Guía la idea de cómo soportar bajas dependencias entre las clases, bajo impacto del cambio e incrementar la reutilización. La respuesta sería asignar una responsabilidad de manera que el acoplamiento permanezca bajo (Larman, 2003). Es una medida de la fuerza con que un elemento está conectado o tiene conocimiento de otros elementos. Este patrón es utilizado por el framework Django, debido a que utiliza las URL para separar clases y funcionalidades, esto permite que si deja de funcionar alguna de las vistas no afecta a las demás.

Alta cohesión:

Guía la idea de cómo mantener una complejidad manejable. La respuesta sería asignar una responsabilidad de manera que la cohesión permanezca alta. La cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas y que no hace una gran cantidad de trabajo, tiene alta cohesión (Larman, 2003). Se evidencia en la clase ApacheServerAdmin, puesto que maneja solo la información asociada a los servicios Apache.

Controlador

Guía la idea de quien debe ser el responsable de gestionar un evento de entrada al sistema. La respuesta sería asignar la responsabilidad de recibir o manejar un mensaje de evento al sistema a una clase que representa el sistema global, dispositivo o subsistema (Larman, 2003). Se evidencia en el archivo views.py, porque es el encargado de definir las clases y funcionalidades que controlan las vistas que se muestran en el sistema.

2.8 Tarjetas Clase Responsabilidad Colaboración (CRC)

La utilización de tarjetas CRC (*Class-Responsibility-Collaboration*, por sus siglas en inglés) es una técnica de diseño orientado a objetos propuesta por Kent Beck (introducido de la metodología de programación extrema). Identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades). Cada tarjeta contiene el nombre de la clase que representa una o más historias de usuario, una descripción de las responsabilidades o métodos asociados con la clase. Además, contiene una lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP (Quintero, 2014).

Tabla 14 Tarjeta CRC No.1

Tarjeta CRC	
Clase: ApacheServerAdmin	
Responsabilidad	Colaboración
Se encarga de realizar las funciones sobre el servicio Apache.	Librería Paramiko ApacheStatus CreateApache2Group Servidor

Tabla 15 Tarjeta CRC No.2

Tarjeta CRC	
Clase: ApacheStatus	
Responsabilidad	Colaboración
Se encarga de monitorear los recursos asignados al servicio Apache.	Librería pandas

Tabla 16 Tarjeta CRC No.3

Tarjeta CRC	
Clase: RemoteUsers	
Responsabilidad	Colaboración
Esta clase se encarga de la gestión de usuario en los servidores remotos que tendrán o no permiso de edición en los ficheros de configuración de Apache.	Servidor

Tabla 17 Tarjeta CRC No.4

Tarjeta CRC	
Clase: Servidor	
Responsabilidad	Colaboración
Esta clase se encarga de definir los datos almacenados para la gestión de servidores remotos.	Models

Conclusiones del capítulo

En el presente capítulo se realizó una descripción de la solución propuesta, permitiendo definir las principales funcionalidades del sistema. Se definieron 13 historias de usuario implementadas en un plan de 3 iteraciones que se obtuvo en la fase de planificación, lo cual permitió describir los aspectos principales para el desarrollo de la solución propuesta. Como parte del diseño del sistema, se definió el patrón arquitectónico, el cual se tomó como base para la propuesta de solución. Se realizaron las tarjetas CRC, las cuales representan las clases del sistema, relaciones entre ellas y sus funcionalidades. Se identificaron los patrones de diseño, lo que permitió facilitar la implementación, la asignación de responsabilidades y proporcionando mayor independencia entre las clases.

Capítulo 3: Implementación y pruebas

Introducción

Para lograr que un producto de software sea de calidad adecuada es necesario realizar determinadas actividades de manera sistemática. En el presente capítulo, se define la etapa de implementación que consiste en el desarrollo y programación del código, teniendo en cuenta los estándares que propone Python para una mejor organización del mismo. Se describen las tareas de ingeniería que se generan como artefactos de la metodología XP. Posteriormente se realiza la fase de pruebas, para comprobar la calidad del sistema implementado.

3.1 Implementación

Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de alto nivel, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo. XP promueve la programación basada en estándares, de manera que sea entendible por todo el equipo, y que facilite la recodificación.

3.1.1 Estándares de codificación

XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. Al comenzar un software, se establece un estándar de codificación para asegurar que todos los programadores del proyecto trabajen de forma coordinada. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Basabe, 2012).

Para la definición de las clases y métodos en el código del sistema y basándose en la guía de estilo Pep8 para código Python, fueron utilizados los siguientes estándares de codificación (Foundation, 2013).

- No se deben utilizar tildes en el código y los comentarios se realizan utilizando el símbolo de número y el comentario seguido de este.
- CapWords: Cuando las palabras se inician con mayúscula de manera continua. Ejemplo: InicialesMayusculas, véase en la figura 12.

- lower_case_with_underscores: Cuando las palabras se escriben en minúscula y separadas por guión bajo. Ejemplo: guion_bajo, véase en la figura 13.
- Las importaciones deben estar en líneas separadas, por ejemplo:

```
#!/ -*- coding: utf-8 -*-
import io
import os
import re
import string
import random
import time
import paramiko
import pandas as pd
import as np
```

Figura 11 Importaciones

```
class ServidorForm(forms.ModelForm):
    class Meta:
        model = Servidor
        fields = ['nombre', 'ip', 'port', 'os', 'user', 'password']
        widgets = {
            'password': forms.PasswordInput,
        }

class VirtualHostForm(forms.Form):
    file_name = forms.CharField(max_length=255, label='Nombre de fichero')
    root_dir = forms.CharField(max_length=255, initial='/var/www/html/', label='Directorio raiz')
    host_name = forms.CharField(max_length=255, label='Nombre del host')
    alias = forms.CharField(required=False, widget=forms.Textarea(attrs={'rows': 10})),
```

Figura 12 CapWords

```
if editing is False:
    ctx = {
        'server': {
            'address_ports': [line.split(':') for line in vhf.cleaned_data['addr_port']]
            |
            'name': vhf.cleaned_data['host_name'],
            'admin': vhf.cleaned_data['server_admin'],
            'document_root': vhf.cleaned_data['root_dir'],
            'error_log': vhf.cleaned_data['error_log'],
            'use_ssl': vhf.cleaned_data['use_ssl'],
            'ssl_cert': vhf.cleaned_data['ssl_cert'],
            'ssl_priv_key': vhf.cleaned_data['ssl_priv_key'],
            'alias': [line for line in vhf.cleaned_data['alias'].splitlines()],
        }
    }
```

Figura 13 lower_case_with_underscores

3.1.2 Tareas de Ingeniería

Son el artefacto que se genera en la metodología XP en la fase de implementación. Se elaboran para ayudar y simplificar la programación de cada una de las HU. Una HU puede tener más de una tarea de ingeniería lo cual depende de la complejidad de dicha HU. A continuación, se definen las Tareas de Ingeniería correspondientes a cada una de las HU especificadas en el capítulo anterior. Las restantes tareas de ingeniería se evidencian en el **Anexo 3**.

Tabla 18 Tarea de ingeniería “Autenticar usuario”

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Autenticar usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 10 de marzo de 2018	Fecha de fin: 10 de marzo de 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: Va a permitir al administrador introducir el nombre de usuario y contraseña para autenticarse en el sistema. Si está mal algún parámetro se mostrará un mensaje con el error correspondiente, sino se muestra la página de inicio.	

Tabla 19 Tarea de ingeniería “Establecer conexión por SSH a un servidor remoto”

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 3
Nombre Tarea: Establecer conexión por SSH a un servidor remoto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 25 de marzo de 2018	Fecha de fin: 25 de marzo de 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: El sistema le va a permitir al administrador establecer conexiones por SSH a los servidores remotos. Si la conexión se realizó correctamente se mostrará la página de inicio del sistema. En caso de fallar la conexión se mostrará un mensaje de error.	

Tabla 20 Tarea de ingeniería “Instalar servicio Apache”

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 4

Nombre Tarea: Instalar servicio Apache	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 26 de marzo de 2018	Fecha de fin: 26 de marzo de 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: El administrador podrá instalar el servicio Apache en un servidor remoto. Se le notificará con un mensaje al concluir la instalación. En caso de que el servicio ya esté instalado en el servidor se mostrará un mensaje correspondiente, sino mostrará un mensaje de error.	

Tabla 21 Tarea de ingeniería “Cambiar estado del servicio Apache”

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 5
Nombre Tarea: Cambiar estado del servicio Apache	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha de inicio: 27 de marzo de 2018	Fecha de fin: 30 marzo de 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: El sistema le permitirá al administrador manejar los estados del servicio Apache: iniciar, detener, reiniciar y recargar dicho servicio. Al seleccionar un estado se mostrará un mensaje de notificación correspondiente.	

Tabla 22 Tarea de ingeniería “Desinstalar servicio Apache”

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 6
Nombre Tarea: Desinstalar servicio Apache	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha de inicio: 31 de marzo de 2018	Fecha de fin: 1 de abril de 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: Brinda al administrador la opción de desinstalar el servicio Apache en un servidor remoto. En caso de que el servicio no se encuentre en el servidor mostrará un mensaje correspondiente. Si ocurre algún error se mostrará un mensaje.	

Tabla 23 Tarea de ingeniería “Mostrar Logs”

Tarea de Ingeniería	
Número Tarea: 17	Número Historia de Usuario: 11
Nombre Tarea: Mostrar Logs	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 26 de abril de 2018	Fecha de fin: 26 de abril de 2018
Programador Responsable: Emilio Gort Caballero	
<p>Descripción: El sistema le va a permitir al administrador la opción de mostrar información sobre los Logs de accesos y de errores de Apache. Al ocurrir algún error se le notificará con un mensaje.</p>	

Tabla 24 Tarea de ingeniería “Monitorear servicio Apache”

Tarea de Ingeniería	
Número Tarea: 18	Número Historia de Usuario: 12
Nombre Tarea: Monitorear servicio Apache	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 27 de abril de 2018	Fecha de fin: 27 de abril de 2018
Programador Responsable: Emilio Gort Caballero	
<p>Descripción: Esta funcionalidad le va a permitir al administrador mostrar información de los recursos del servidor utilizados por el servicio Apache como CPU y memoria RAM, así como la cantidad de bytes leídos y escritos por cada proceso del servicio Apache.</p>	

Tabla 25 Tarea de ingeniería “Adicionar hosts virtuales”

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 8
Nombre Tarea: Adicionar hosts virtuales	
Tipo de Tarea: Gestión	Puntos Estimados: 0.2
Fecha de inicio: 7 abril de 2018	Fecha de fin: 7 de abril de 2018
Programador Responsable: Emilio Gort Caballero	
<p>Descripción: El administrador podrá al seleccionar un servicio Apache remoto, adicionar hosts virtuales, introduciendo los datos especificados por el mismo en un formulario. Si algún parámetro está mal se le notificará con un mensaje de error. Al</p>	

adicionarse un host virtual se mostrará un mensaje correspondiente. En caso de existir algún error de configuración se mostrará la dirección del error en el estado de configuración del servicio Apache.

Tabla 26 Tarea de ingeniería “Eliminar hosts virtuales”

Tarea de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: 8
Nombre Tarea: Eliminar hosts virtuales	
Tipo de Tarea: Gestión	Puntos Estimados: 0.2
Fecha de inicio: 13 de abril 2018	Fecha de fin: 13 de abril 2018
Programador Responsable: Emilio Gort Caballero	
Descripción: El sistema le permitirá al administrador eliminar un host virtual. Se mostrará un mensaje de confirmación al eliminarse dicho host virtual. Al ocurrir algún error se le notificará en un mensaje.	

3.2 Pruebas del software

Las pruebas son una de las prácticas fundamentales en las cuales se basa XP. Esta actividad se realiza de forma continua a lo largo del proyecto, pues se requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Existen dos tipos de pruebas, las unitarias y las de aceptación. Las pruebas unitarias son definidas por los programadores antes de comenzar a escribir el código. Las pruebas de aceptación se definen para cada historia de usuario al principio de cada iteración, para validar que cada requerimiento implementado funciona como se había especificado (Vargas, 2015).

3.2.1 Pruebas unitarias

Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Permiten tener conocimiento en tiempo real sobre la respuesta del sistema. Una prueba de unidad pretende saber si cada función en un archivo de programa simple funciona correctamente, ayudando a independizar un módulo y facilitando la posibilidad de realizar una prueba independientemente del resto del sistema (Gómez Palomo, et al., 2014).

En la figura 14 se muestra el resultado de las pruebas unitarias que se realizaron a través del módulo que ofrece Python llamado PyUnit. Se le realizaron pruebas a la clase modelo Servidor y la clase ApacheServerAdmin a las funcionalidades: `install_apache`, `is_apache_installed`, `reload_apache`,

restart_apache, start_apache, stop_apache, uninstall_apache, enable_virtual_host, delete_vh_file, create_apache2_group y apache_logs.

```

Creating test database for alias 'default'...
.....
-----
Ran 16 tests in 0.030s

OK
Destroying test database for alias 'default'...

```

Figura 14 Resultado de la ejecución de las pruebas unitarias utilizando el módulo PyUnit

3.2.2 Pruebas de aceptación

Las pruebas de aceptación son de vital importancia para el éxito de una iteración y el comienzo de la siguiente, con lo cual el cliente puede conocer el avance en el desarrollo del sistema. Además, permite una retroalimentación para el desarrollo de las próximas HU a ser entregadas. Una HU no es aceptada hasta que haya pasado su prueba de aceptación. Estas son comúnmente llamadas pruebas del cliente, por lo que son realizadas por el encargado de verificar si las HU de cada iteración cumplen con la funcionalidad esperada (Meléndez, 2016). Las demás pruebas de Aceptación pueden consultarse en el Anexo 4.

Tabla 27 Prueba de aceptación “Autenticar usuario”

Caso de prueba de aceptación	
Código: HU_P1	Historia de Usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad de autenticar usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: Se intenta autenticar un usuario en el sistema con los datos válidos.	
Resultados esperados: El usuario se autentica correctamente en el sistema.	

Tabla 28 Prueba de aceptación “Establecer conexión por SSH a un servidor remoto”

Caso de prueba de aceptación	
Código: HU_P8	Historia de Usuario: 3

Nombre: Establecer conexión por SSH a un servidor remoto.
Descripción: Prueba para la funcionalidad de establecer conexión por SSH a un servidor remoto.
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servidor remoto debe estar registrado en el sistema. El servidor remoto debe tener configurado el servidor SSH para establecer la conexión.
Pasos de ejecución: El usuario se autentica en el sistema. Se selecciona el servidor remoto en la lista de servidores.
Resultados esperados: La conexión se realiza satisfactoriamente.

Tabla 29 Prueba de aceptación "Instalar servicio Apache"

Caso de prueba de aceptación	
Código: HU_P4	Historia de Usuario: 4
Nombre: Instalar servicio Apache	
Descripción: Prueba para la funcionalidad de instalar servicio Apache en un servidor remoto.	
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servicio Apache no puede estar instalado en el servidor remoto.	
Pasos de ejecución: Se accede al sistema con las credenciales correctas. Se selecciona el servidor remoto donde se vaya a instalar el servicio Apache. Se escoge la opción instalar. Se muestra un mensaje: "El servidor Apache ha sido instalado satisfactoriamente".	
Resultados esperados: El servicio Apache se instala satisfactoriamente.	

Tabla 30 Prueba de aceptación "Cambiar estado del servicio Apache"

Caso de prueba de aceptación	
Código: HU_P9	Historia de Usuario: 5

Nombre: Cambiar estado del servicio Apache
Descripción: Prueba para la funcionalidad de cambiar estado del servicio Apache en un servidor remoto.
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto.
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servidor remoto donde se vaya a cambiar el estado del servicio Apache. Escoger la opción que se desea por el usuario para cambiar el estado del servicio Apache remoto. Se muestra un mensaje: "Operación realizada con éxito". Oprimir el botón Cerrar.
Resultados esperados: Se realizan los cambios de estado del servicio Apache satisfactoriamente.

Tabla 31 Prueba de aceptación "Eliminar Host Virtual"

Caso de prueba de aceptación	
Código: HU_P17	Historia de Usuario: 8
Nombre: Eliminar Host Virtual.	
Descripción: Prueba para la funcionalidad de eliminar host virtual.	
Condiciones de ejecución: El usuario tiene que estar autenticado como administrador en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto. El servidor remoto debe tener instalado los protocolos OpenSSH.	
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servicio Apache remoto. Acceder a la funcionalidad Hosts virtuales. Se muestra un mensaje: "Desea eliminar este fichero de configuración virtual host?" Oprimir el botón aceptar. Se muestra un mensaje: "El fichero ha sido eliminado satisfactoriamente"	
Resultados esperados:	

Se elimina un host virtual satisfactoriamente.

Tabla 32 Prueba de aceptación “Habilitar y deshabilitar Host Virtual”

Caso de prueba de aceptación	
Código: HU_P16	Historia de Usuario: 8
Nombre: Habilitar y deshabilitar host virtual.	
Descripción: Prueba para la funcionalidad de habilitar y deshabilitar host virtual.	
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto. El servidor remoto debe tener instalado los protocolos SSH y SSL.	
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servicio Apache remoto. Acceder a la funcionalidad Hosts virtuales. En la opción estado, dar clic sobre el botón para cambiar el estado del host virtual seleccionado. Oprimir el botón aceptar, para verificar que se desea cambiar el estado del host virtual.	
Resultados esperados: Se muestra el cambio de estado de un host virtual.	

Tabla 33 Prueba de aceptación “Desinstalar servicio Apache”

Caso de prueba de aceptación	
Código: HU_P10	Historia de Usuario: 6
Nombre: Desinstalar servicio Apache	
Descripción: Prueba para la funcionalidad de desinstalar servicio Apache en un servidor remoto.	
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto.	
Pasos de ejecución: Se accede al sistema con las credenciales correctas. Se selecciona el servidor remoto donde se vaya a desinstalar el servicio Apache.	

<p>Escoger la opción desinstalar.</p> <p>Seleccionar la opción aceptar para verificar que se desea desinstalar Apache.</p> <p>Se muestra un mensaje: “Operación realizada con éxito”.</p> <p>Oprimir el botón Cerrar.</p>
<p>Resultados esperados:</p> <p>El servicio Apache se desinstala satisfactoriamente.</p>

Tabla 34 Prueba de aceptación “Restringir acceso al servicio Apache”

Caso de prueba de aceptación	
Código: HU_P11	Historia de Usuario: 7
Nombre: Restringir acceso al servicio Apache	
Descripción: Prueba para la funcionalidad de restringir acceso al servicio Apache.	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado en el sistema.</p> <p>El servicio Apache debe estar instalado en el servidor remoto.</p> <p>El sistema debe estar conectado al servidor remoto.</p>	
<p>Pasos de ejecución:</p> <p>Acceder al sistema con los datos correctos.</p> <p>Seleccionar el servidor remoto donde se vaya a restringir el acceso de usuarios a los archivos de configuración del servicio Apache.</p> <p>Oprimir el botón gestionar usuarios.</p> <p>Oprimir el botón para cambiar los permisos de edición de cada usuario.</p>	
<p>Resultados esperados:</p> <p>Se cambian los permisos de edición del usuario seleccionado.</p>	

Tabla 35 Prueba de aceptación “Modificar puertos para las conexiones en Apache”

Caso de prueba de aceptación	
Código: HU_P18	Historia de Usuario: 9
Nombre: Modificar puertos para las conexiones en Apache.	
Descripción: Prueba para la funcionalidad de modificar puertos para las conexiones en Apache.	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado en el sistema</p>	

El servicio Apache debe estar instalado en el servidor remoto. El servidor debe tener instalado el protocolo SSH.
Pasos de ejecución: Acceder al sistema con las credenciales correctas. Seleccionar un servicio Apache remoto. Acceder a la funcionalidad Puertos de escucha. Escoger la opción editar. Modificar los datos en el formulario. Seleccionar la opción guardar. Ejecutar el botón aceptar.
Resultados esperados: El sistema muestra los datos modificados de los puertos de escucha.

Tabla 36 Prueba de aceptación “Mostrar Logs”

Caso de prueba de aceptación	
Código: HU_P22	Historia de Usuario: 11
Nombre: Mostrar Logs.	
Descripción: Prueba para la funcionalidad de mostrar Logs.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema. El servicio Apache debe estar instalado en el servidor remoto. El servidor debe tener instalado el protocolo SSH.	
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servicio Apache remoto. Escoger la funcionalidad Logs. Seleccionar la opción Logs de acceso o Logs de error.	
Resultados esperados: Se muestra un listado con los Logs de accesos y otro listado con los Logs de errores satisfactoriamente.	

Tabla 37 Prueba de aceptación “Monitorear servicio Apache”

Caso de prueba de aceptación	
Código: HU_P23	Historia de Usuario: 12

Nombre: Monitorear servicio Apache.
Descripción: Prueba para la funcionalidad de monitorear servicio Apache.
Condiciones de ejecución: El usuario debe estar autenticado en el sistema. El servicio Apache debe estar instalado en el servidor remoto. El servidor debe tener instalado el protocolo SSH.
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servicio Apache remoto.
Resultados esperados: El sistema muestra gráficas con los recursos asignados al servicio Apache.

Tabla 38 Prueba de aceptación “Adicionar usuario”

Caso de prueba de aceptación	
Código: HU_P24	Historia de Usuario: 13
Nombre: Adicionar usuario.	
Descripción: Prueba para la funcionalidad de adicionar usuario.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto. El servidor debe tener instalado el protocolo SSH.	
Pasos de ejecución: Acceder al sistema con los datos válidos. Acceder al módulo Usuarios. Oprimir el botón adicionar usuario. Insertar los datos requeridos en el formulario. Oprimir el botón Adicionar del formulario.	
Resultados esperados: El sistema muestra la adición de un usuario en la tabla.	

Tabla 39 Prueba de aceptación “Eliminar usuario”

Caso de prueba de aceptación	
Código: HU_P29	Historia de Usuario: 13

Nombre: Eliminar usuario.
Descripción: Prueba para la funcionalidad de eliminar usuario.
Condiciones de ejecución: El usuario debe estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto. El servidor debe tener instalado el protocolo SSH.
Pasos de ejecución: Oprimir el botón eliminar del usuario seleccionado. Se muestra un mensaje: “Realmente quiere eliminar el usuario?” Seleccionar la opción: Sí
Resultados esperados: Se elimina el usuario satisfactoriamente.

Tabla 40 Prueba de aceptación “Editar Host Virtual”

Caso de prueba de aceptación	
Código: HU_P14	Historia de Usuario: 8
Nombre: Editar Host Virtual.	
Descripción: Prueba para la funcionalidad de editar host virtual.	
Condiciones de ejecución: El usuario tiene que estar autenticado en el sistema. El servicio Apache tiene que estar instalado en el servidor remoto. El servidor remoto debe tener instalado los protocolos SSH y SSL.	
Pasos de ejecución: Acceder al sistema con los datos válidos. Seleccionar el servicio Apache remoto. Acceder a la funcionalidad Hosts virtuales. Dar clic en el nombre del host que se desea modificar. Realizar los cambios requeridos en el formulario: Editar Host Virtual. Seleccionar la opción guardar.	
Resultados esperados: El sistema muestra el host virtual con los datos modificados satisfactoriamente.	

Las pruebas de aceptación fueron realizadas a las historias de usuario, presentando en la primera iteración 4 no conformidades las cuales fueron resueltas. En la segunda iteración se encontraron 2 no

conformidades las cuales fueron corregidas y la tercera iteración arrojó 1 no conformidad la cual fue rectificadas en su totalidad. A continuación, en la siguiente tabla se especifican las no conformidades por iteración.

Tabla 41 “No conformidades identificadas por iteraciones”

No.	No Conformidades	Iteración
1	La eliminación de máquinas servidoras en el listado sin una previa confirmación de esta.	1
2	La no validación de los datos al adicionar o editar una máquina servidora.	1
3	La no confirmación al usuario al instalar el servicio Apache en un servidor remoto.	1
4	La ocurrencia de errores al establecer conexión con una máquina remota sin notificar al usuario.	1
5	Permanencia de los datos de información una vez desinstalado el servicio Apache.	2
6	La no validación de los nombres de ficheros de los hosts virtuales sin o con el sufijo: “. conf”.	2
7	La no realización del filtrado de datos o información específica sobre los Logs de acceso y el fichero de errores.	3

A continuación, se muestra una gráfica con los resultados alcanzados luego de realizar las pruebas de aceptación al software.

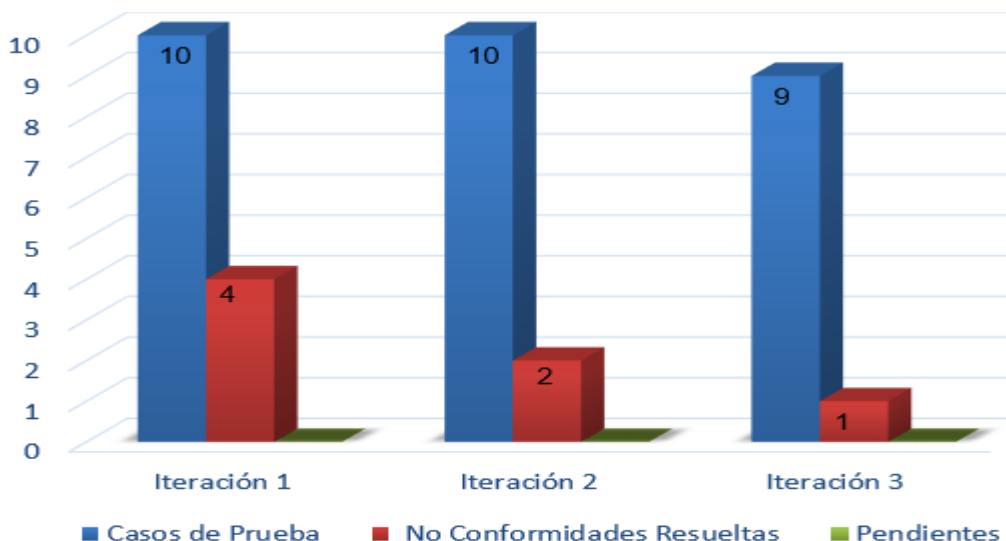


Figura 15 Resultados de la ejecución de pruebas de aceptación

3.4 Conclusiones del capítulo

En el presente capítulo se definió el estándar de codificación que abarca la estructura que tendrá el código del sistema, permitiendo una mejor comprensión por parte del equipo de desarrollo. Las pruebas realizadas permitieron determinar el correcto funcionamiento del mismo, garantizando el cumplimiento de las necesidades del cliente reflejadas en las historias de usuario. Además, se implementaron las tareas de ingeniería, las cuales favorecieron una descripción de las tareas realizadas para implementar las funcionalidades y estimar el tiempo de duración para el desarrollo del sistema propuesto.

Conclusiones

Al finalizar el presente trabajo de diploma, se cumple con el objetivo general planteado al inicio de la investigación, por lo que se puede arribar a las siguientes conclusiones:

1. Se realizó un análisis del estado del arte de sistemas homólogos existentes. Esto permitió identificar la necesidad de desarrollar un sistema para la gestión, administración y monitorización de servicios web Apache para el Departamento de Componentes del Centro de TLM, puesto que ninguno satisfacía las necesidades específicas del área.
2. La selección de las herramientas informáticas y lenguajes de programación permitió el desarrollo de un sistema acorde con las necesidades establecidas por el cliente. Se realizó un estudio sobre las diferentes metodologías de desarrollo de software, seleccionándose XP para guiar el proceso de implementación del sistema.
3. Se realizó una descripción del proceso que se lleva a cabo en el Departamento de Componentes del Centro de TLM, permitiendo un mejor entendimiento del negocio y el diseño de una solución para dar cumplimiento a las necesidades del área.
4. El sistema fue desarrollado haciendo uso de los estándares de codificación, garantizando la limpieza y organización en el código.
5. Se realizaron pruebas definidas por la metodología de desarrollo de software XP para la validación de la calidad y el correcto funcionamiento del sistema con resultados satisfactorios.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en esta investigación se recomienda:

1. Extender el sistema GAMSW en cualquier entidad que requiera su uso para gestionar, administrar y monitorear servicios web Apache.
2. Añadir para próximas versiones del sistema nuevas funcionalidades como: Analizar Logs y Optimizar el servicio web Apache.

Glosario de términos

TLM Telemática.

GAMSW Gestion, Administración, Monitorización de Servidores Web.

TIC Tecnologías de la Información y las Comunicaciones.

HTTP Protocolo de Transferencia de Hipertexto.

HTTPS Protocolo Seguro de Transferencia de Hipertexto.

GUI Interfaz Gráfica de Usuario.

IP Internet Protocol.

SSI Server Side Includes.

SSL Capa de Conexión Segura.

URI Identificador uniforme de recursos.

XML Lenguaje de Marcado Extensible.

BIND Dominio de Nombre de Internet de Berkeley.

DNS Sistema de Nombres de Dominio.

CVS Sistema de Versiones Concurrentes.

DHCP Dynamic Host Configuration Protocol.

FTP File Transfer Protocol.

HMAST Herramienta para la Migración y Administración de Servicios Telemáticos.

XP Programación Extrema.

HTML5 Lenguaje de Marcado de Hipertexto.

CSS3 Hojas de Estilo en Cascada.

OOCSS CCS Orientado a Objetos.

CASE Ingeniería de Software Asistida por Computadora.

BPMN Business Process Model and Notation.

IDE Entorno de Desarrollo Integrado.

SGBD Sistema Gestor de Base de Datos.

HU Historias de Usuario.

GRASP Patrones de Software de Asignación de Responsabilidad General.

CPU Unidad Central de Procesamiento.

RAM Memoria de Acceso Aleatorio.

PHP Hypertext Preprocessor.

CRC Tarjetas Clase Responsabilidad Colaboración.

AJAX Asynchronous JavaScript and XML.

Referencias Bibliográficas

Aldas Mena, Daniel Ernesto y Andrade Cadena, Maritza Alejandra. 2011. *Guía práctica para el uso de patrones de diseño en el desarrollo de software.* 2011.

Alejandra Torres. 2017. ComparaHosting. [En línea] 2 de octubre de 2017. [Citado el: 4 de octubre de 2017.] <https://www.comparahosting.io/que-es-un-servidor/>.

Ambuludí. Científico, Artículo. 2014. 2014.

Apache. 2017. Versión 2.2 de la documentación del Servidor de HTTP Apache. [En línea] 2017. [Citado el: 4 de octubre de 2017.] <http://httpd.apache.org/docs/2.2/es/>.

ApacheGUI. 2015. Apache GUI. [En línea] 2015. [Citado el: 8 de noviembre de 2017.] <http://www.apachegui.net/>.

Belloch, Consuelo. aprendizaje, Las Tecnologías de la Información y Comunicación en el. 2012. 2012, INFORMACION COMERCIAL ESPANOLA-MONTHLY EDITION-, Vol. 25, pág. 125.

bizagi. 2014. *BPMN.* 2014.

Bootstrap. 2015. Apuntes de Programación. [En línea] 4 de mayo de 2015. [Citado el: 11 de octubre de 2017.] <http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>.

Castillo, Alberto Azoye. 2017. *Curso de Programacion Web JavaScript, AJAX y JQuery.* 2017.

. 2017. *Curso de Programación Web: JavaScript, Ajax y jQuery.* IT Campus Academy. 2017.

CCM. 2016. . *CCM. Patrones de Diseño.* [En línea] 21 de octubre de 2016. [Citado el: 1 de marzo de 2018.] es.ccm.net.

Chavez, Erick Ruiz de. 2013. desarrolloweb.com. [En línea] 2 de julio de 2013. [Citado el: 11 de octubre de 2017.] <https://desarrolloweb.com/articulos/que-es-backbonejs.html>.

Ciencia, tecnología, Sociedad e Innovacion. **María Pilar Martínez Ruiz. 2006.** Valencia : UOC, 2006.

Cobo, Ángel. 2007. *Diseño y programación de bases de datos.* s.l. : Visión Libros, 2007.

Contreras Ortiz, Fabio Alberto. 2017. *dispositivos moviles que permita el pago de servicios publicos de manera segura y confiable.* BOGOTA : s.n., 2017.

Damian, Bhrunis Flores Jeffry. 2017. *LAS TIC'S Y SU INCIDENCIA EN EL DESARROLLO DE LAS DESTREZAS CON CRITERIO DE DESEMPEÑO EN LOS ESTUDIANTES DE LA UNIDAD EDUCATIVA.* Ecuador : s.n., 2017.

Django. 2016. BBVA Open4U. [En línea] 2016 de enero de 2016. [Citado el: 11 de octubre de 2017.] <https://bbvaopen4u.com/es/actualidad/django-guia-rapida-para-desarrollar-paginas-web-con-este-framework>.

Falcón Márquez, Odalys Rosa. 2016. UCI. [En línea] 21 de junio de 2016. [Citado el: 15 de febrero de 2018.] <http://www.uci.cu/la-uci-y-la-informatizacion-de-la-sociedad-cubana>.

García Collazo, Antonio. 2015. *Facultad.* 2015.

- Gauchat, Juan Diego. 2012.** *El gran libro HTML5, JavaScript, CSS3.* 2012.
- Gómez Palomo, Sebastián Rubén y Moraleda Gil, Eduardo. 2014.** *Aproximación a la ingeniería del software.* España : Editorial Universitaria Ramon Areces, 2014. ISBN-13:978-84-9961-093.
- González Tolmo, Débora . Médica, Sistema de teleconsulta médica. Revista Cubana de Informática. 2012.* 1, 15 de noviembre de 2012, Vol. 12.
- González, Rolando Alfredo Hernández León y Sayda Coello. 2012.** *El proceso de Investigación Científica.* 2 Edición. La Habana : Editorial Universitaria, 2012.
- González, Sayda Coello. 2012.** *El proceso de Investigación Científica.* 2 Edición. La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2012.
- Gregorio, Jose. 2011.** Servidores Web. [En línea] 28 de junio de 2011. [Citado el: 10 de octubre de 2017.] dis.um.es/~lopezquesada/documentos/IES_1213/IAW/curso/UT4/.../2/index.html.
- Guizado Freire, Elina Katherine. 2013.** *Reestructuración del código de la Librería Ajax de consumo de servicios Web a través del protocolo Soap para el mejoramiento de la adaptabilidad en el desarrollo de sistemas dentro de la infraestructura de Besixplus Cia. Ltda.* 2013.
- Hernández Dominguez, Antonio. 2018.** UCI. [En línea] 2018. [Citado el: 15 de febrero de 2018.] <http://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-telematica-tlm>.
- Hernández, Jesús. 2014.** *Análisis y Desarrollo Web.* 2014.
- Hinojosa Tinoco, Diego Andrés. 2013.** *Arquitectura De Software.* Ecuador : s.n., 2013.
- IDE. 2015.** Joomla! Documentation. [En línea] 25 de agosto de 2015. [Citado el: 11 de octubre de 2017.] [https://docs.joomla.org/Category:IDE_\(Integrated_development_environment\)/es](https://docs.joomla.org/Category:IDE_(Integrated_development_environment)/es).
- Infante Montero, Sergio. 2012.** *Maestros del Web.* Eugenia Tobar. 2012.
- JetBRAINS. 2017.** PyCharm. [En línea] 2017. [Citado el: 12 de octubre de 2017.] <https://www.jetbrains.com/pycharm/features/>.
- JM Pantoja Blyde, A Lozano Leal. 2013.** Télématique. [En línea] 2013. [Citado el: 10 de octubre de 2017.] <http://publicaciones.urbe.edu/index.php/telematique/article/view/2306/pdf>.
- josegregoriob. 2011.** Servidor Web. [En línea] 28 de junio de 2011. [Citado el: 13 de febrero de 2018.] <https://es.slideshare.net/josegregoriob/servidor-web-8451426>.
- Joskowicz, José. 2008.** *Reglas y Prácticas en eXtreme Programming.* España : s.n., 2008.
- kabir, Mohammed J. 2004.** *La_Biblia_Server_apache.* 2004.
- Kennethe. kendall, Julie E. Kendall. 2011.** *Análisis y diseño de sistemas.* [ed.] Luis Miguel Cruz Castillo. OCTAVA EDICIÓN,. New Jersey : s.n., 2011. ISBN: 978-607-32-0577-1.

- Mendoza, Dayana. 2016.** *Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso.* La Habana : s.n., 2016. 2306-2495.
- Larman, Craig. 2003.** *UML y Patrones.* segunda. Madrid : s.n., 2003. 84-205-3438-2.
- Letelier, Patricio. 2006.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] 2006. [Citado el: 21 de febrero de 2018.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.
- . 2006.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* 2006.
- ManageEngine. 2017.** ManageEngine . [En línea] 2017. [Citado el: 24 de octubre de 2017.] <https://www.manageengine.com/>.
- Maribel Monzó. 2016.** Todo Turia. [En línea] 13 de julio de 2016. todotur.com/el-impacto-de-las-tic-en-la-sociedad-actual/.
- Martínez, Pedro Pablo Fábrega. 2013.** Configuración y Administración de Apache. [En línea] 2013. [Citado el: 5 de octubre de 2017.] <http://www.bdat.net/documentos/apache/>.
- Visual Paradigm. 2018.** Visual Paradigm. [En línea] 2018. [Citado el: 2018 de mayo de 2.] <https://www.visual-paradigm.com/>.
- Meléndez Valladarez, Sinthya Milena. 2016.** *MELÉNDEZ VALLADAREZ, Sinthya Milena, et al. Sistema WEB de evaluación al desempeño Docente UNAN-Managua, empleando la metodología Agil Programación Extrema.* Nicaragua : s.n., 2016.
- Miranda . ferriarcos, Business intelligence para manejo y recuperación de cartera en la empresa. 2012.** Ecuador : s.n., 2012.
- Molina, Juan Carlos y Torres, M. E. 2010.** *Análisis de requerimientos usando BPMN.* Colombia : s.n., 2010. págs. 85-97. Vol. 11. 1.
- Monsalve Galeano, Elizabeth. 2015.** *UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA. PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES.* 2015.
- Montoya Suarez, Lina Maria. 2017.** *Análisis comparativo de las metodologías ágiles en el desarrollo de software aplicadas en Colombia.* Colombia : s.n., 2017.
- Nubity. 2014.** Nubity. [En línea] 2014. [Citado el: 27 de octubre de 2017.] <https://nubity.com>.
- Pérez Ríos, Cynthia Katterine. 2014.** *La calidad del servicio al cliente y su influencia en los resultados económicos y financieros de la empresa Restaurante Campestre SAC, Chiclayo periodo enero a setiembre 2011 y 2012.* 2014.
- Pérez, Javier Eguíluz. 2009.** *Introducción a CSS.* 2009.
- . 2009.** *Introducción a HTML.* 2009.
- Quintero. extrema, Desarrollo ágil de software aplicando programación. 2014.** 1, 2014, Revista Ingenio UFPSO, Vol. 5, págs. 24-29.

Rendón Basabe, Gabriela. 2012. *Análisis, diseño, desarrollo e implementación de un sistema de venta de boletos de cine para smartphones utilizando Visual Studio.* 2012.

Rodríguez, Leover Armando González. 2011. Alternativas para el desarrollo de Aplicaciones Web. [En línea] junio de 2011. [Citado el: 11 de octubre de 2017.] <http://www.informatica-juridica.com/trabajos/alternativas-para-el-desarrollo-de-aplicaciones-web/>.

Romani, Susana. 2013. Catalogación como Apoyo al Uso de Patrones de Seguridad. 5to Workshop de Seguridad Informatica, WSegl. Santa Fe Argentina. : s.n., 2013.

Romero, Hermes. 2012. Metodologías de Desarrollo. [En línea] 7 de febrero de 2012. [Citado el: 10 de octubre de 2017.] <https://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>.

Rossum, Guido van. 2017. *Tutorial de Python.* [ed.] Jr. Fred L. Drake. 2017.

SISTEMA DE MONITOREO Y CONTROL DEL MÓDULO DE ANÁLISIS DE SECUENCIAS DE LA PLATAFORMA DE SERVICIOS BIOINFORMÁTICOS. **Navarro Rosquete, Diana, Bussott Pérez, Briseida y Díaz León, Andry Daniel. 2015.** Ciudad de La Habana : s.n., 2015.

SQLite. 2018. SQLite. [En línea] de de 2018. [Citado el: 24 de abril de 2018.] <https://www.sqlite.org>.

Vargas Almendáriz, Luis Miguel. 2015. *Metodología de desarrollo de software dirigida a equipos de trabajo reducidos para su aplicación en los proyectos integradores y tesis en Uniandes Extensión Ibarra.* 2015.

Vila Alvarez, Arley. 2018. *Desarrollo de una plataforma web para la gestión de contratos de prestación de servicios y paz y salvos para el servicio nacional de aprendizaje sena.* 2018.

Webmin. 2016. Webmin. [En línea] 2016. [Citado el: 24 de octubre de 2017.] <http://www.webmin.com/>.

yosandra Sandoval, Georgina Gallizo. 2012. Evaluation of monitoring tools for cloud computing environments. En Informática(CLEI). [En línea] 2012. [Citado el: 24 de octubre de 2017.] <https://www.researchgate.net/publication/261300349>.

Foundation, Python Software. 2013. Python. *Style Guide for Python Code.* [Online] 8 1, 2013. [Cited: abril 4, 2018.] <https://www.python.org/dev/peps/pep-0008/>.

Bibliografía

- Aldas Mena, Daniel Ernesto and Andrade Cadena, Maritza Alejandra. 2011.** *Guía práctica para el uso de patrones de diseño en el desarrollo de software.* 2011.
- Alejandra Torres. 2017.** ComparaHosting. [Online] octubre 2, 2017. [Cited: octubre 4, 2017.] <https://www.comparahosting.io/que-es-un-servidor/>.
- Ambuludí. Científico, Artículo. 2014.** 2014.
- Apache. 2017.** Versión 2.2 de la documentación del Servidor de HTTP Apache. [Online] 2017. [Cited: octubre 4, 2017.] <http://httpd.apache.org/docs/2.2/es/>.
- ApacheGUI. 2015.** Apache GUI. [Online] 2015. [Cited: noviembre 8, 2017.] <http://www.apachegui.net/>.
- bizagi. 2014.** *BPMN.* 2014.
- Bootstrap. 2015.** Apuntes de Programación. [Online] mayo 4, 2015. [Cited: octubre 11, 2017.] <http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>.
- Castillo, Alberto Azoye. 2017.** *Curso de Programación Web JavaScript, AJAX y JQuery.* 2017.
- . 2017.** *Curso de Programación Web: JavaScript, Ajax y jQuery.* IT Campus Academy. 2017.
- CCM. 2016.** . *CCM. Patrones de Diseño.* [Online] octubre 21, 2016. [Cited: marzo 1, 2018.] es.ccm.net.
- Chavez, Erick Ruiz de. 2013.** desarrolloweb.com. [Online] julio 2, 2013. [Cited: octubre 11, 2017.] <https://desarrolloweb.com/articulos/que-es-backbonejs.html>.
- Cobo, Ángel. 2007.** *Diseño y programación de bases de datos.* s.l. : Visión Libros, 2007.
- Consuelo. aprendizaje, Las Tecnologías de la Información y Comunicación en el. 2012.** 2012, INFORMACION COMERCIAL ESPANOLA-MONTHLY EDITION-, Vol. 25, p. 125.
- Contreras Ortiz, Fabio Alberto. 2017.** *dispositivos móviles que permita el pago de servicios públicos de manera segura y confiable.* BOGOTA : s.n., 2017.
- Damian, Bhrunis Flores Jeffrey. 2017.** *LAS TIC'S Y SU INCIDENCIA EN EL DESARROLLO DE LAS DESTREZAS CON CRITERIO DE DESEMPEÑO EN LOS ESTUDIANTES DE LA UNIDAD EDUCATIVA.* Ecuador : s.n., 2017.
- Django. 2016.** BBVA Open4U. [Online] enero 2016, 2016. [Cited: octubre 11, 2017.] <https://bbvaopen4u.com/es/actualidad/django-guia-rapida-para-desarrollar-paginas-web-con-este-framework>.
- Falcón Márquez, Odalys Rosa. 2016.** UCI. [Online] junio 21, 2016. [Cited: febrero 15, 2018.] <http://www.uci.cu/la-uci-y-la-informatizacion-de-la-sociedad-cubana>.
- Fernández, Arturo. 2013.** *Python 3 al descubierto.* 2013. ISBN.
- García Collazo, Antonio. 2015.** *Facultad.* 2015.

Gauchat, Juan Diego. 2012. *El gran libro de HTML5, CSS3 y Javascript*. Marcombo. Barcelona : s.n., 2012. ISBN eBook: 978-84-267-1782-5.

. 2012. *El gran libro HTML5, JvaScript, CSS3*. 2012.

Gómez Palomo, Sebastián Rubén and Moraleda Gil, Eduardo. 2014. *Aproximación a la ingeniería del software*. España : Editorial Universitaria Ramon Areces, 2014. ISBN-13:978-84-9961-093.

González . Médica, Sistema de teleconsulta médica. Revista Cubana de Informática. 2012. 1, noviembre 15, 2012, Vol. 12.

González, Rolando Alfredo Hernández León y Sayda Coello. 2012. *El proceso de Investigación Científica*. 2 Edición. La Habana : Editorial Universitaria, 2012.

González, Sayda Coello. 2012. *El proceso de Investigación Científica*. 2 Edición. La Habana : Editorial Universitaria del Ministerio de Educación Superior, 2012.

Gregorio, Jose. 2011. Servidores Web. [Online] junio 28, 2011. [Cited: octubre 10, 2017.] dis.um.es/~lopezquesada/documentos/IES_1213/IAW/curso/UT4/.../2/index.html.

Guizado Freire, Elina Katherine. 2013. *Reestructuración del código de la Librería Ajax de consumo de servicios Web a través del protocolo Soap para el mejoramiento de la adaptabilidad en el desarrollo de sistemas dentro de la infraestructura de Besixplus Cia. Ltda.* 2013.

Hernández Dominguez, Antonio. 2018. UCI. [Online] 2018. [Cited: febrero 15, 2018.] <http://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-telematica-tlm>.

Hernández, Jesús. 2014. *Análisis y Desarrollo Web*. 2014.

Hinojosa Tinoco, Diego Andrés. 2013. *Arquitectura De Software*. Ecuador : s.n., 2013.

IDE. 2015. Joomla! Documentation. [Online] agosto 25, 2015. [Cited: octubre 11, 2017.] [https://docs.joomla.org/Category:IDE_\(Integrated_development_environment\)/es](https://docs.joomla.org/Category:IDE_(Integrated_development_environment)/es).

Infante Montero, Sergio. 2012. *Maestros del Web*. Eugenia Tobar. 2012.

JetBRAINS. 2017. PyCharm. [Online] 2017. [Cited: octubre 12, 2017.] <https://www.jetbrains.com/pycharm/features/>.

JM Pantoja Blyde, A Lozano Leal. 2013. Télématique. [Online] 2013. [Cited: octubre 10, 2017.] <http://publicaciones.urbe.edu/index.php/telematique/article/view/2306/pdf>.

josegregoriob. 2011. Servidor Web. [Online] junio 28, 2011. [Cited: febrero 13, 2018.] <https://es.slideshare.net/josegregoriob/servidor-web-8451426>.

Joskowicz, José. 2008. *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.

kabir, Mohammed J. 2004. *La_Biblia_Server_apache*. 2004.

- Kennethe kendall, Julie E. Kendall. 2011.** *Análisis y diseño de sistemas*. [ed.] Luis Miguel Cruz Castillo. OCTAVA EDICIÓN,. New Jersey : s.n., 2011. ISBN: 978-607-32-0577-1.
- Larman, Craig. 2003.** *UML y Patrones*. segunda. Madrid : s.n., 2003. 84-205-3438-2.
- Letelier, Patricio. 2006.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [Online] 2006. [Cited: febrero 21, 2018.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.
- . 2006.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2006.
- MAGISTER, SANDRO. 2018.** Paramiko. [Online] 2018. [Cited: abril 2018, 3.] <http://www.paramiko.org/>.
- ManageEngine. 2017.** ManageEngine . [Online] 2017. [Cited: octubre 24, 2017.] <https://www.manageengine.com/>.
- Maribel Monzó. 2016.** Todo Turia. [Online] julio 13, 2016. todotur.com/el-impacto-de-las-tic-en-la-sociedad-actual/.
- Martínez, Pedro Pablo Fábrega. 2013.** Configuración y Administración de Apache. [Online] 2013. [Cited: octubre 5, 2017.] <http://www.bdat.net/documentos/apache/>.
- Meléndez Valladares, Sinthya Milena. 2016.** MELÉNDEZ VALLADAREZ, Sinthya Milena, et al. *Sistema WEB de evaluación al desempeño Docente UNAN-Managua, empleando la metodología Agil Programación Extrema*. Nicaragua : s.n., 2016.
- Miranda . ferriarcos, Business intelligence para manejo y recuperación de cartera en la empresa. 2012.** Ecuador : s.n., 2012.
- Molina, Juan Carlos and Torres, M. E. 2010.** *Análisis de requerimientos usando BPMN*. Colombia : s.n., 2010. pp. 85-97. Vol. 11. 1.
- Monsalve Galeano, Elizabeth. 2015.** UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA. PROGRAMA DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES. 2015.
- Montoya Suarez, Lina Maria. 2017.** *Análisis comparativo de las metodologías ágiles en el desarrollo de software aplicadas en Colombia*. Colombia : s.n., 2017.
- Nubity. 2014.** Nubity. [Online] 2014. [Cited: octubre 27, 2017.] <https://nubity.com>.
- Pérez Ríos, Cynthia Katterine. 2014.** *La calidad del servicio al cliente y su influencia en los resultados económicos y financieros de la empresa Restaurante Campestre SAC, Chiclayo periodo enero a setiembre 2011 y 2012*. 2014.
- Pressman, Roger. 2010.** *Ingeniería del Software. Un Enfoque Práctico*. séptima. s.l. : MC GRAW HILL, 2010. p. 777. ISBN.
- Quintero. extrema, Desarrollo ágil de software aplicando programación. 2014.** 1, 2014, Revista Ingenio UFPSO, Vol. 5, pp. 24-29.
- Rendón Basabe, Gabriela. 2012.** *Análisis, diseño, desarrollo e implementación de un sistema de venta de boletos de cine para smartphones utilizando Visual Studio*. 2012.

Rodríguez, Leover Armando González. 2011. Alternativas para el desarrollo de Aplicaciones Web. [Online] junio 2011. [Cited: octubre 11, 2017.] <http://www.informatica-juridica.com/trabajos/alternativas-para-el-desarrollo-de-aplicaciones-web/>.

Romero, Hermes. 2012. Metodologías de Desarrollo. [Online] febrero 7, 2012. [Cited: octubre 10, 2017.] <https://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>.

Rossum, Guido van. 2017. *Tutorial de Python*. [ed.] Jr. Fred L. Drake. 2017.

Seguridad en la configuración del servidor web Apache. **Gómez, Carlos Eduardo. 2013.** 2, 2013, Vol. 9, pp. 31-38.

SISTEMA DE MONITOREO Y CONTROL DEL MÓDULO DE ANÁLISIS DE SECUENCIAS DE LA PLATAFORMA DE SERVICIOS BIOINFORMÁTICOS. **Navarro Rosquete, Diana, Bussott Pérez, Briseida and Díaz León, Andry Daniel. 2015.** Ciudad de La Habana : s.n., 2015.

Sommerville, Ian. 2005. *Ingeniería del software*. [trans.] María Isabel Alfonso Galipienso. Addison Wesley. s.l. : Pearson Educación, 2005. ISBN.

SQLite. 2018. SQLite. [Online] , 2018. [Cited: abril 24, 2018.] <https://www.sqlite.org>.

Vargas Almendáriz, Luis Miguel. 2015. *Metodología de desarrollo de software dirigida a equipos de trabajo reducidos para su aplicación en los proyectos integradores y tesis en Uniandes Extensión Ibarra*. 2015.

Vila Alvarez, Arley. 2018. *DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE CONTRATOS DE PRESTACIÓN DE SERVICIOS Y PAZ Y SALVOS PARA EL SERVICIO NACIONAL DE APRENDIZAJE SENA*. 2018.

Webmin. 2016. Webmin. [Online] 2016. [Cited: octubre 24, 2017.] <http://www.webmin.com/>.

Foundation, Python Software. 2013. Python. *Style Guide for Python Code*. [Online] 8 1, 2013. [Cited: abril 4, 2018.] <https://www.python.org/dev/peps/pep-0008/>.