

**UNIVERSIDAD JOSÉ ANTONIO PÁEZ**

**DESARROLLO DE UN PANEL DE CONTROL** **WEB PARA LA ADMINISTRACIÓN DE SERVIDORES**

**Y CLÚSTERES CON SISTEMA OPERATIVO GNU/LINUX CASO**

**RED HAT ENTERPRISE LINUX 6**

**Autor:**

González, Irisel

20.313.217

Urb. Yuma II, calle No 3. Municipio San Diego

Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394

**REPÚBLICA BOLIVARIANA DE VENEZUELA**

**UNIVERSIDAD JOSÉ ANTONIO PÁEZ**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA EN COMPUTACIÓN**

**DESARROLLO DE UN PANEL DE CONTROL WEB PARA LA ADMINISTRACIÓN DE SERVIDORES Y CLÚSTERES CON SISTEMA OPERATIVO GNU/LINUX CASO RED HAT ENTERPRISE LINUX 6**

**Trabajo de Grado presentado como requisito para optar al título de**

**INGENIERO DE COMPUTACIÓN**

**Autora:** Irisel Alejandra González Alcalá

C.I. 20.313.217

**Tutor:** Ing. Luis Llave

San Diego, Mayo 2013



REPÚBLICA BOLIVARIANA DE VENEZUELA

UNIVERSIDAD JOSÉ ANTONIO PÁEZ

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

**ACEPTACIÓN DEL TUTOR**

Quien suscribe, Ingeniero Luis Llave portador de la cédula de identidad N° 15.864.521, en mi carácter de tutor del trabajo de grado presentado por la ciudadana, portadora de la cédula de identidad N° 20.313.217, titulado **DESARROLLO DE UN PANEL DE CONTROL WEB PARA LA ADMINISTRACIÓN DE SERVIDORES Y CLÚSTERES CON SISTEMA OPERATIVO GNU/LINUX CASO RED HAT ENTERPRISE LINUX 6** presentado como requisito parcial para optar al título de Ingeniero de Computación, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los 21 días del mes de mayo del año dos mil trece.

Ing. Luis Llave.

C.I.: 15.864.521

REPÚBLICA BOLIVARIANA DE VENEZUELA

UNIVERSIDAD JOSÉ ANTONIO PÁEZ

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

San Diego, Mayo 2013

**ACTA DE REVISIÓN DEL PROYECTO DE TRABAJO DE GRADO**

Quienes suscriben esta Acta, dejan constancia que el Proyecto de Trabajo de Grado:

**DESARROLLO DE UN PANEL DE CONTROL WEB PARA LA ADMINISTRACIÓN DE SERVIDORES Y CLÚSTERES CON SISTEMA OPERATIVO GNU/LINUX CASO RED HAT ENTERPRISE LINUX 6** , ha sido revisado y, cumpliendo con los requisitos exigidos para su aprobación, recomiendan su tramitación ante el organismo académico correspondiente.

Ing. Luis Llave \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Tutor Académico Firma Fecha

Ing. Alicia Pizzella \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Tutor Metodológico Firma Fecha

**ÍNDICE GENERAL**

CONTENIDO Pp.

**RESUMEN**………………………………….……………………………………..

**INTRODUCCIÓN**………………………………….……………………………...

**CAPÍTULO**

**I EL PROBLEMA**

1.1 Planteamiento del Problema………………………………….………….

1.2 Formulación del Problema……………………….……………………....

1.3 Objetivos de la Investigación………..…………………………………...

1.3.1 Objetivo General……………..............................................................

1.3.2 Objetivos Específicos………………………………………………..

1.4 Justificación………………………………………………........................

1.5 Alcance…………………………………………………………………..

**II MARCO TEÓRICO**

2.1 Antecedentes…………………………………………………………….

2.2 Bases Teóricas……………………………………………………………

2.2.1 Servidores…………………………………………………………….

2.2.1.1 Servidor de Correo………….……………………………………..

2.2.1.2 Servidor de Fax………….…………………………………………

2.2.1.3 Servidor Proxy………….………………………………………….

2.2.1.4 Servidor de acceso remoto (RAS)……………….............................

2.2.1.5 Servidor de uso………….………………………….........................

2.2.1.6 Servidor Web………….…………………………............................

2.2.1.7 Servidor HTTP Apache………….………………..………...............

2.2.1.8 Servidor DNS………….……………………….....………...............

2.2.1.9 Servidor de Base de Datos………….…………….………................

2.2.1.10 Servidor de Archivo………….………………….………................

2.2.1.11 Servidor FTP………….……….................………............................

2.2.1.12 Servidor de DHCP………….……………………………................

2.2.1.13 Servidor de OpenSSH………….………………..……….................

2.2.1.14 Servidor de Reserva………….………………….……….................

2.2.1.15 Servidor de Seguridad………….………………..……….................

2.2.1.16 Servidor Dedicado………….……………………………................

2.2.1.17 Servidor no Dedicado………….………………...………................

2.2.2 Clústeres……………………………………………….………................

2.2.2.1 Clúster de alta disponibilidad……………………...………................

2.2.2.2 Cluster de alta eficiencia………….……………………….................

2.2.2.3 Balanceo de carga………….……………………..……….................

2.2.3 Git………………………………………………………………………

2.2.4 Lenguaje de Programación…………………………….…………….

2.2.5 Lenguaje Etiqueta……………………………………..…………….

2.2.6 HTML5……………………………………………….…………….

2.2.7 CSS3………………………………………………………………..

2.2.8 Less…………………………………………………..……………..

2.2.9 Twitter Bootstrap……………………………………..…………….

2.2.10 Framework…………………………………………..…………….

2.2.11 Flask……………………………………………………………….

2.2.12 NoSQL……………………………………………...……………..

2.2.12.1 Base de Datos documentales…………………………………….

2.2.12.1 Base de Datos en Grafos……………………….…………….....

2.2.12.1 Base de Datos Clave-Valor……………………..……………....

2.2.12.1 Base de Datos Tabulares………………………...……………...

2.2.13 MongoDB………………………………………………………….

2.3 Definición de Términos…………………………………..………………

**III MARCO METODOLOGICO**

3.1 Tipo de Investigación

3.2 Diseño de la Investigación………………………………………………

3.3 Nivel de la Investigación………………………………………………..

3.4 Fases Metodológicas...…………………………………….…………….

**IV RECURSOS**

4.1 Humanos…………………………………………………..…………….

4.2 Institucionales……………………………………………..…………….

4.3 Materiales………………………………………………….…………….

**V REFERENCIAS BIBLIOGRAFICAS**……………………..…………….

REPÚBLICA BOLIVARIANA DE VENEZUELA

UNIVERSIDAD JOSÉ ANTONIO PÁEZ

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

**DESARROLLO DE UN PANEL DE CONTROL WEB PARA LA ADMINISTRACIÓN DE SERVIDORES Y CLÚSTERES CON SISTEMA OPERATIVO GNU/LINUX CASO RED HAT ENTERPRISE LINUX 6**

**Autora:** Irisel Alejandra González Alcalá

**Tutor:** Ing. Luis Llave

**Fecha:** Mayo, 2013

**RESUMEN INFORMATIVO**

La investigación plantea el desarrollo de un panel de control web para la administración de Servidores y Clústeres con sistema operativo GNU/Linux caso Red Hat Enterprise Linux 6, donde se formula la interrogante: ¿Cómo optimizar la administración y configuración de Servidores y Clústeres para facilitar la interacción y gestión de los servicios?, que tiene por objetivos, realizar la escritura y ejecución de pruebas, codificar los distintos módulos del sistema, realizar la Refactorización (Refactoring) o Reestructuración, especificar las pruebas de Aceptación y generar la documentación necesaria. El trabajo está bajo la modalidad de tipo documental y factible ya que es una propuesta viable destinada a satisfacer necesidades específicas a partir de un diagnóstico. Apoyada en una investigación de campo, según el manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales de la Universidad Pedagógica Experimental Libertador (UPEL, 2006). Este proyecto se apoya en una investigación de tipo descriptivo, documental y de campo en la modalidad de proyecto factible. Además se realizará a través de fases metodológicas de Desarrollo Guiado a Pruebas (TTD), llamada también Test-Driven Developmen siendo la Fase I: Escritura y ejecución de pruebas, Fase II: Escritura de código, Fase III: Refactorización (Refactoring) o Reestructuración y Fase IV: Especificación de pruebas de Aceptación. Se contará con factores externos como apoyo que consisten en un conjunto de personas, bienes materiales, financieros y técnicos, para el desarrollo de la investigación, guiada por un cronograma de actividades para lograr con éxito la culminación de la misma.

**Descriptores:** Sistema operativo-Linux-Servidor-Clúster

**INTRODUCCIÓN**

Hoy día, la tecnología computacional ha generado un estilo propio en la sociedad, creando cambios de forma de vida, costumbres y hasta la forma de pensar, al vivir en un mundo globalizado impuesto por las grandes potencias. La aparición de Linux (sistema operativo de código abierto), se ha apoderado lentamente de la cultura informática del mundo empresarial, creando un adelanto computacional.

Linux Torvalds creador original del sistema Linux en la Universidad de Helsinki en Finlandia, inicio su trabajo con simples ideas para un núcleo de un sistema operativo, que giraban en torno al desarrollo de un pequeño sistema UNIX de carácter académico dirigido a aquellos usuarios de Minix que querían algo más. El desarrollo inicial de Linux ya aprovechaba las características de conmutación de tareas en modo protegido del microprocesador Intel 80386.

En Octubre de 1991, Linux anunció la primera versión "oficial" de Linux, la 0.02. Ya podía ejecutar bash (el shell de GNU) y gcc (el compilador de C de GNU), pero no hacía mucho más. Luego alcanzó la versión 0.95, y la versión "oficial" en Marzo de 1992. Año y medio después, en Diciembre de 1993, el núcleo estaba en la revisión 0.99, en una aproximación asintótica al 1.0. Actualmente, el núcleo se encuentra en la versión 3.8.6 y 3.9-rc6 lanzada el 5 y 8 de Abril de 2013.

A la fecha, hay cientos de desarrolladores de software de todo el mundo que contribuyen al esfuerzo de software Linux; debido a que el código fuente del software está disponible gratuitamente, cualquier persona puede trabajar en él, cambiarlo o mejorarlo, los desarrolladores están encargados de contribuir en sus correcciones y mejoras en la comunidad para que Linux pueda seguir creciendo y mejorando, pero su valor no radica en las piezas, sino en sus características más relevantes, ya que los equipos con software Linux, son económicos, flexibles, estables, de fácil administración, expandibles y adaptables, para solucionar los problemas de los clientes.

En consecuencia tomando como referencia el motivo del presente estudio el cual es, el desarrollo de un panel de control web para la administración de servidores y clústeres con sistema operativo gnu/linux caso red hat enterprise linux 6, es de señalar, que en los últimos años han nacido proyectos como son los paneles de control, software que provee una interfaz gráfica para la administración de los servicios del servidor, brindando un gran nivel de control al usuario, manteniendo la seguridad; siendo mayormente en sistemas operativos basados en Linux.

En base a estos planteamientos, la presente investigación se estructura en cuatro capítulos desarrollados de la siguiente manera: en el capítulo I, se delinea el planteamiento y formulación del problema, la justificación de la investigación y el alcance, donde se analiza cada uno de estos aspectos en detalle para darle vialidad a la investigación.

El capítulo II, constituye el marco teórico conceptual en el cual se presentan los antecedentes de la investigación, que hacen referencia a la temática planteada, las bases teóricas y definición de términos, que expresan la situación general de la investigación.

El capítulo III contiene el marco metodológico, donde se hace referencia al tipo, diseño y fases de la investigación que conforman el trabajo.

En lo que respecta al capítulo IV, lo conforma los recursos utilizados y el cronograma de las diversas etapas de presentación del proyecto.

Finalmente, las referencias utilizadas impresas y electrónicas donde se extrajo la información teórica y de apoyo para el desarrollo de la investigación.

**CAPÍTULO I**

**EL PROBLEMA**

* 1. **Planteamiento del Problema**

En la actualidad la tecnología computacional ha marcado un auge significativo en la sociedad, tanto para el individuo como para el mundo de las organizaciones, cambiando así la forma de vida, costumbres y forma de pensar del individuo, a tal punto que en la actualidad, algunas personas se resisten a vivir fuera del sistema globalizado que imponen las grandes potencias. La aparición de Linux que en sus principios pintaba como un pequeño sistema operativo de código abierto, se ha apoderado poco a poco de la cultura informática a nivel mundial, entrado de lleno en el mundo empresarial, generando así un gran avance computacional.

Linux fue creado originalmente por Linux Torvalds, de 21 años, en la Universidad de Helsinki en Finlandia. En abril de 1991, empezó a trabajar en unas simples ideas para un núcleo de un sistema operativo, inicialmente, sólo fue un proyecto de aficionado de Linux Torvalds, el cual se inspiraba en Minix, un pequeño UNIX desarrollado por Andy Tanenbaum, y las primeras discusiones sobre Linux surgieron en el grupo de noticias comp.os.minix. Estas discusiones giraban en torno al desarrollo de un pequeño sistema UNIX de carácter académico dirigido a aquellos usuarios de Minix que querían algo más. El desarrollo inicial de Linux ya aprovechaba las características de conmutación de tareas en modo protegido del microprocesador Intel 80386, y se escribió todo en ensamblador, Linux expresó:

"Comencé a utilizar el C tras escribir algunos controladores de dispositivos, y ciertamente se aceleró el desarrollo. En este punto sentí que mi idea de hacer "un Minix mejor que Minix" se hacía más seria. Esperaba que algún día pudiese recompilar el gcc bajo Linux... "Dos meses de trabajo, hasta que tuve un controlador de dispositivo de discos con numerosos errores de software, pero que parecía funcionar en mi PC y un pequeño sistema de ficheros. Aquí tenía ya la versión 0.01 [al final de Agosto de 1991], no era muy agradable de usar sin el controlador de dispositivos de disquetes, y no hacía gran cosa. No pensé que alguien compilaría esa versión."

Tras numerosas revisiones, se alcanzó la versión 0.95, reflejando la esperanza de tener lista muy pronto una versión "oficial", esto sucedía en Marzo de 1992. Año y medio después, en Diciembre de 1993, el núcleo estaba en la revisión 0.99, en una aproximación asintótica al 1.0. Actualmente, el núcleo se encuentra en la versión 3.8.6 y 3.9-rc6 lanzada el 5 y 8 de Abril de 2013.

Hoy en día, hay cientos de desarrolladores de software de todo el mundo que contribuyen al esfuerzo de software Linux; debido a que el código fuente del software está disponible gratuitamente, cualquier persona puede trabajar en él, cambiarlo o mejorarlo, los desarrolladores están encargados de contribuir en sus correcciones y mejoras en la comunidad para que Linux pueda seguir creciendo y mejorando, pero su valor no radica en las piezas, sino en sus características más relevantes, ya que los equipos con software Linux, son económicos, flexibles, estables, de fácil administración, expandibles y adaptables, para solucionar los problemas de los clientes.

Linux se centró en mantener la comunicación abierta entre los desarrolladores de software, creciendo en una cultura de libre intercambio de ideas de software. La adquisición del código para trabajar era el objetivo, sin preocuparse mucho acerca de quién era dueño del código, e Internet fue el medio principal de comunicaciones. El núcleo de Linux está legalmente protegido por la Licencia Pública GNU (GPL de sus siglas en inglés), se establecía como un buen camino para compartir software, sin embargo para que Linux fuera aceptado en una pequeña y técnica población computacional, era necesario que fuera fácil de instalar y usar. Igualmente empresarios que piensan en cumplir sus objetivos críticos, aplicándolo en un sistema de computadoras, quieren saber si este sistema fue probado cuidadosamente.

Para esos fines muchas compañías y organizaciones decidieron empacar juntos software de Linux en unas formas usables llamadas “distribuciones”. La meta principal de una distribución de Linux es hacer cientos de paquetes de software no relacionados que hacen que Linux trabaje unido como un todo cohesivo, satisfaciendo las necesidades de un grupo específico de usuarios. En los últimos años la distribución comercial más popular ha sido Red Hat Linux.

Red Hat es una distribución Linux creada por Red Hat Inc, en Septiembre de 2003 Red Hat Inc, ha cambiado la manera de hacer negocio, con la distribución Red Hat Enterprise Linux, también conocida por sus siglas RHEL. Es la versión comercial basada en el anterior Red Hat Linux, con este cambio también se forma el proyecto Fedora, patrocinado por Red Hat, para tomar el desarrollo de la tecnología de Red Hat Linux en el futuro.

Red Hat Enterprise Linux, es un producto Linux oficial y comercial de Red Hat Inc, siendo un sistema operativo de código abierto con flexibilidad, eficiencia y control, lo que lo convierte en la solución a través de una amplia gama de arquitectura de hardware, monitores de máquina virtual y acceso a los servicios a través del internet.

En el mundo empresarial, las organizaciones necesitan mucho poder de cálculo y almacenamiento disponible las 24 horas los 7 días de la semana, debido a la gran demanda de procesos que requiere, por lo tanto un servidor, que no es más que un conjunto de hardware y software conectados en una red que realiza una serie de tareas en beneficio de un cliente o usuario, le proporciona gran ayuda y fiabilidad a la organización.

Los servidores son parte fundamental de una organización, ya que ayuda a reforzar su productividad y rentabilidad, proteger sus sistemas y sus datos, brindándole mayor eficiencia, incrementando la velocidad y confiabilidad de los procesos. Un servidor para su funcionamiento necesita de un software operativo, que soporte las necesidades y exigencias del usuario, y uno de los más populares son los basados en el núcleo Linux. Red Hat Enterprise Linux, considerada la mejor y más utilizada plataforma para servidores empresariales en la actualidad, esta distribución brinda innovaciones tecnológicas, altos niveles de fiabilidad, escalabilidad, rendimiento y seguridad, además de proporcionar valor a los clientes y mayor flexibilidad en la infraestructura.

La plataforma cuenta con una serie de certificaciones profesionales dirigidas a reforzar y garantizar las habilidades técnicas de la administración de sistemas a los operadores en sus responsabilidades profesionales. Las certificaciones basadas en Red Hat Enterprise Linux, van desde Implementación de Sistemas de Gestión: Red Hat Enterprise, Gestión de Clúster y almacenamiento en Red Hat, Virtualización Red Hat Enterprise, Red Hat Certified Engineer (RHEC), entre otras, siendo esta última una de las más conocidas y común. Estas certificaciones ayudaran al administrador de sistemas aprender a implementar y administrar Red Hat Enterprise de manera eficiente y eficaz, así como gestionar almacenamiento compartidos y agrupamiento de servidores, adquiriendo experiencia en la herramienta de gestión de almacenamiento Red Hat Clúster Suite, como también el adquirir las habilidades necesarias para implantar y gestionar tecnologías de Virtualización Red Hat Enterprise, siéndole de gran ayuda en el crecimiento de su vida profesional.

A pesar del gran auge y reconocimiento de la plataforma, las organizaciones requieren uno o más administradores de sistemas para poder obtener los resultados deseados. Esto obliga a requerir personal calificado para el mantenimiento y administración de cada uno de estos servidores. Lo cual se ha convertido en uno de los grandes problemas durante años, por la cantidad de profesionales con pocos conocimientos sólidos en la distribución Red Hat Enterprise Linux, lo que ha llevado a las organizaciones a migrar a otras distribuciones u otros sistemas operativos. Además de la complejidad y tedioso que conlleva la configuración, tanto del servidor como de los servicios que presta, ya que la mayoría de estas configuraciones se hacen mediante la codificación, que requiere mayor tiempo y a veces genera complicaciones.

En los últimos años han nacido proyectos para ayudar a la administración de servidores, como lo son los paneles de control, que es un software que provee una interfaz gráfica para la administración de los servicios del servidor, brindando un gran nivel de control al usuario, manteniendo la seguridad; siendo mayormente en sistemas operativos basados en Linux, esto ha originado que muchas organizaciones estén utilizando e implementando el sistema operativo GNU/Linux; pero a pesar de esto no se puede gestionar los procesos básicos y necesarios de servidores y clústeres de una manera más ágil y optima, que ofrezca beneficio, fiabilidad, seguridad, además, de la gestión de recursos de una red.

**1.2. Formulación del Problema**

¿Cómo optimizar la administración y configuración de Servidores y Clústeres para facilitar la interacción y gestión de los servicios?.

**1.3. Objetivos de la Investigación**

**1.3.1. Objetivo General**

Desarrollar un panel de control web para la administración de Servidores y Clústeres con sistemas operativo GNU/Linux caso: Red Hat Enterprise Linux 6.

* + 1. **Objetivos Específicos**
* Realizar la escritura y ejecución de pruebas.
* Codificar los distintos módulos del sistema.
* Realizar la Refactorización (Refactoring) o Reestructuración.
* Especificar las pruebas de Aceptación
* Generar la documentación necesaria.

**1.4. Justificación**

En los últimos años el Software Libre en Latinoamérica, ha tenido un crecimiento elevado en cuanto a su uso, no solo en usuarios particulares, sino en las grandes empresas, ya que un porcentaje de las organizaciones del sector público y privado hacen un gran uso del software libre en parte de sus aplicaciones, incluyendo servidores web, sistemas operativos de servidor, navegadores web, base de datos, entre otros. Pero a pesar de su uso e implementación de aplicaciones de servidores en las organizaciones, carecen de un panel para la administración de servidores y clústeres óptimos bajo Software Libre en español que ofrezca beneficio, fiabilidad y seguridad, además, de la gestión de recursos de una red.

Por esta razón, se busca con el desarrollo del software facilitar al administrador de sistemas, un panel que le permita gestionar servidores y clústeres de una manera más rápida y óptima, ya que el software contará con una interfaz sencilla y amigable para su utilización, sin necesidad de la codificación a la hora de configurar los servicios, simplemente teniendo los conocimientos y dominios básicos en el área.

**1.5. Alcance de la Investigación**

Con la realización de este proyecto, se pretende abarcar las funciones que cumple el panel de administración de servidores y clústeres, la metodología que se aplicará para el desarrollo de determinado software y las herramientas necesarias para alcanzar el objetivo deseado. Ahora bien, el software contará con una interfaz para administrar los siguientes servicios: FTP, HTTP, DNS, SSH, BD, EMAIL (correos).

La metodología a utilizar en esta investigación, será Desarrollo Guiado a Pruebas (TTD), la cual consiste en primer lugar, a diseñar y escribir los casos de prueba y las pruebas unitarias en base a requisitos del software y a la arquitectura del proyecto, en segundo lugar, se codificará la aplicación de tal manera que el código implementado cumpla los test diseñados previamente; garantizando la reutilización de código, una interfaz más clara y un código más limpio y eficiente.

Las herramientas a utilizar para la codificación y desarrollo del software serán: el lenguaje Python, framework Cherrypy y sistema de base datos no relacional MongoDB.

**CAPÍTULO II**

**MARCO TEÓRICO**

**2.1. Antecedentes**

Iturriaga S., Maya P., Pintos D. (2008), en su trabajo de grado titulado **Proyecto Fenton - Clúster de Computadores de Alto Desempeño con Acceso Remoto (CCADAR).** Para optar al título de Ingeniero en Computación en la Universidad de la República Uruguay. El proyecto de tesis consistió en la implementación de un clúster de computadoras de alto desempeño con la capacidad de brindar acceso remoto para que sus usuarios interactúen con él y para que sus administradores realicen monitoreo y regulación.

Con el proyecto el grupo responsable del clúster desea mantener un control estricto de las actividades y recursos por parte de los distintos usuarios y procesos; tanto para poder realizar un seguimiento de sus estado actual y verificar su correcto funcionamiento, como para realizar mediciones referidas a la utilización de recursos por parte de los usuarios. Así como también permitir el monitoreo del cluster en tiempo real por parte de los administradores.

Mediante este trabajo de grado se obtuvo conocimientos con respecto a la evaluación y diseño de herramientas automáticas para administración y utilización de un clúster de computadoras.

Por otra parte, Rocío V. Laura R., Huezo P. Martha E., Parada R. Octavio R., (2009), en su trabajo de grado titulado **Análisis del Sistema Operativo Gnu/Linux. Para optar por el título de Licenciado en Ciencias de la Computación.** En su proyecto de grado, plantean, el estudio del direccionamiento de la memoria desde la óptica de los microprocesadores Intel, en este caso la familia 80x86. Aquí es notable la importancia que tiene la arquitectura del microprocesador sobre la manera en que se direcciona la memoria y la forma en la cual tiene que ser administrada. Por lo cual, se estudia distintos elementos proporcionados por el microprocesador liderados por la unidad de segmentación y la unidad de paginación; unidades que tienen un papel protagónico en el direccionamiento de la memoria ya que el sistema operativo depende de ellas para poder emplear buenas técnicas de administración de la memoria.

Gracias a este trabajo de grado se obtuvo conocimientos que ayudaron a comprender el funcionamiento de las unidades de segmentación y paginación, y del estudio de cómo el kernel del sistema operativo administra la memoria, comenzando por la administración de marcos de página y todas las funciones y macros relacionadas para poder realizar su correcta asignación y liberación.

A su vez, Montesinos P., Morales José (2010), en su trabajo de grado titulado **“Desarrollo de una Aplicación Web para la Administración de Servidores Proxy en Sistemas Operativos Gnu/Linux. caso: Squid Proxy”** realizado en la Universidad José Antonio Páez, plantean el desarrollo de una herramienta bajo software libre que permita la configuración, administración y el control estadístico de la herramienta Squid Proxy, con una interfaz amigable a usuarios de Tecnología de la Información, disponible a la hora de configurar y administrar la herramienta y proporcionando de manera inmediata estadísticas gráficas que revelen el integro uso de las demandas Web, orientado a todo tipo de empresas u organizaciones.

Mediante este trabajo de grado se obtuvo conocimientos con respecto al desarrollo y manejo de herramientas bajo software libre, las cuales son de gran utilidad aportándole soporte a la solución planteada en el desarrollo de la aplicación, ya que demuestran que un panel de control es necesario para una administración más cómoda de los servicios.

Por último, Meza A. (2012), en su investigación denominada **Análisis de Tráfico de Datos en Sistemas Distribuidos. Para optar al título de Maestro en Ciencias Computacionales del Instituto Tecnológico de la Paz,** abordó el análisis de tráfico en un clúster, utilizando algoritmos genéticos y optimización por enjambre de partículas. Para tal efecto, se efectuaron experimentos en un clúster compuesto por 16 computadoras, usando la librería PVM y el analizador de protocolos Wireshark.

También se analiza el tráfico de datos en una red distribuida y se compara la eficiencia de tres algoritmos relacionados con la transmisión de paquetes y el tiempo de transmisión (Genético Distribuido con Migración Controlada por el Proceso Maestro, Genético Distribuido con Migración en Anillo y Optimización por Enjambre de Partículas Mediante Vecindarios).

A través de este trabajo de grado se obtuvo conocimientos con respecto al análisis de tráfico en redes, para el diagnóstico y funcionamiento de las mismas y cuantificar el volumen de datos que requieren transferir las aplicaciones que se ejecutan en estos ambientes distribuidos.

**2.2. Bases Teóricas**

**2.2.1. Servidores**

Segun García S, Manuel (2013). Define un servidor como:

“Un ordenador o máquina informática que está al servicio de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a estos, todo tipo de información. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.” (p. 2).

Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora antigua, hasta una máquina sumamente potente. Todo esto depende del uso que se le dé al servidor, por lo cual un servidor también puede ser un proceso que entrega información o sirve a otro proceso.

La arquitectura cliente-servidor es un modelo de aplicación distribuida, donde el remitente de una solicitud es conocido como cliente y al receptor de la solicitud enviada por el cliente se conoce como servidor. Este puede ser un proceso que entrega información o sirve a otro proceso, el modelo cliente-servidor no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador.

Diversas aplicaciones se ejecutan en un entorno cliente-servidor. Pero si se requiere de darle un propósito propio, esto dependiendo de la empresa o institución donde se emplean, ya que un servidor puede desarrollarse de forma específica utilizando para ello programas que permiten que el usuario utilice la computadora como servidor o terminal. Tomando en cuenta que existe una gran variedad de servidores que ofrecen al usuario diferentes servicios:

Servidor de Impresiones

Controla una o más impresoras y acepta trabajos de impresión de otros clientes de la red, poniendo en cola los trabajos de impresión, y realizando la mayoría o todas las otras funciones que en el lugar de trabajo se realizaría para lograr una tarea de impresión si la impresora estuviese conectada directamente con el puerto de impresora del lugar de trabajo.

**2.2.1.1. Servidor de Correo**

Es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes.

**2.2.1.2. Servidor de Fax**

Almacena, envía, recibe, en ruta y realiza otras funciones necesarias para la transmisión, la recepción y la distribución apropiadas de los fax.

**2.2.1.3. Servidor Proxy**

Es el servidor que actúa de intermediario de forma que el servidor que recibe una petición sin conocer quién es el cliente que realmente está detrás de esa petición. Permite administrar el acceso a Internet en una red de computadoras permitiendo o negando el acceso a diferentes sitios Web.

**2.2.1.4. Servidor del acceso remoto (RAS)**

Controla las líneas de módem de los monitores u otros canales de comunicación de la red para que las peticiones conecten con la red de una posición remota, responden llamadas telefónicas entrantes o reconoce la petición de la red y realiza la autentificación necesaria y otros procedimientos necesarios para registrar a un usuario en la red.

**2.2.1.5. Servidor de uso**

Realiza la parte lógica de la informática, aceptando las instrucciones para que se realicen las operaciones del lugar de trabajo y sirviendo los resultados a su vez al sitio de labor, mientras que el lugar de trabajo realiza la interfaz operadora o la lógica de la presentación, que se requiere para trabajar correctamente.

**2.2.1.6. Servidor Web**

Almacena principalmente documentos HTML, imágenes, videos, texto, presentaciones, y en general todo tipo de información. Además se encarga de enviar esta información a los clientes.

**2.2.1.7. Servidor HTTP Apache**

Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, entre otros), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

**2.2.1.8. Servidor DNS**

Los usuarios generalmente no se comunican directamente con el servidor DNS; la resolución de nombres se hace de forma transparente por las aplicaciones del cliente (por ejemplo, navegadores, clientes de correo y otras aplicaciones que usan Internet). Al realizar una petición que requiere una búsqueda de DNS, la petición se envía al servidor DNS local del sistema operativo. El sistema operativo, antes de establecer ninguna comunicación, comprueba si la respuesta se encuentra en la memoria caché. Si es así, sirven la respuesta; en caso contrario, iniciarían la búsqueda de manera recursiva. Una vez encontrada la respuesta, el servidor DNS guardará el resultado en su memoria caché para futuros usos y devuelve el resultado.

La mayoría de usuarios domésticos utilizan como servidor DNS el proporcionado por el proveedor de servicios de Internet. La dirección de estos servidores puede ser configurada de forma manual o automática mediante DHCP. En otros casos, los administradores de red tienen configurados sus propios servidores DNS.

**2.2.1.9. Servidor de Base de Datos**

Provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio.

**2.2.1.10. Servidor de Archivo**

Permite el acceso remoto a archivos almacenados en él o directamente accesibles por este. En principio, cualquier ordenador conectado a una red con un software apropiado, puede funcionar como servidor de archivos. Desde el punto de vista del cliente de un servidor de archivos, la localización de los archivos compartidos es compartida y transparente. Eso quiere decir que, normalmente no hay diferencias perceptibles si un archivo está almacenado en un servidor de archivos remoto o en el disco de la propia máquina.

**2.2.1.11. Servidor FTP**

Es un programa especial que se ejecuta en un servidor conectado normalmente en Internet (aunque puede estar conectado en otros tipos de redes, LAN, WAN, entre otras); la función del mismo es permitir el desplazamiento de datos entre diferentes servidores/ordenadores. Los servidores FTP pueden cifrar los comandos de control entre los clientes del FTP y el servidor, así como los datos del archivo, los datos del archivo se aseguran todavía más con el cifrado público.

**2.2.1.12. Servidor DHCP**

Esun protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

**2.2.1.13. Servidor OpenSSH**

Son los encargados de gestionar el control remoto o transferencia de archivos entre equipos. Cuando aparece una petición de conexión, el servidor OpenSSH establece la conexión correcta dependiendo del tipo de herramienta cliente que está conectándose. Por ejemplo, si el equipo remoto se está conectando con la aplicación cliente ssh, el servidor OpenSSH establecerá una sesión de control remoto tras la autenticación. Si el usuario remoto se conecta al servidor OpenSSH con SCP, el dominio del servidor OpenSSH iniciará una copia segura de archivos entre el servidor y el cliente tras la autenticación. OpenSSH puede usar muchos métodos de autenticación, incluyendo contraseñas planas, claves públicas entre otros.

**2.2.1.14. Servidor de Reserva**

Tiene el software de reserva de la red instalado y tiene cantidades grandes de almacenamiento de la red en discos duros u otras formas del almacenamiento disponibles para que se utilice con el fin de asegurarse de que la pérdida de un servidor principal no afecte a la red. Esta técnica también es denominada clustering.

**2.2.1.15. Servidor de Seguridad**

Tiene software especializado para detener intrusiones maliciosas, normalmente tienen antivirus, antispyware, antimalware, además de contar con cortafuegos redundantes de diversos niveles y/o capas para evitar ataques, los servidores de seguridad varían dependiendo de su utilización e importancia. Sin embargo, de acuerdo al rol que asumen dentro de una red se dividen en:

**2.2.1.16. Servidor Dedicado**

Son aquellos que le dedican toda su potencia a administrar los recursos de la red, es decir, a atender las solicitudes de procesamiento de los clientes.

**2.2.1.17. Servidor no Dedicado**

Son aquellos que no dedican toda su potencia a los clientes, sino también pueden jugar el rol de estaciones de trabajo al procesar solicitudes de un usuario local.

**2.2.2. Clústeres**

Mercedes Sinisterra M., Díaz Henao T., Ruiz López E.(2012). Exponen en su trabajo de investigación **Clúster de balanceo de carga y alta disponibilidad para servicios web y mail.**

En la actualidad, debido a la gran demanda de servicios de internet y a la transferencia de la información de todo tipo, es incuestionable que los sistemas Informáticos deben funcionar de forma ininterrumpida y sin errores los 365 días del año. Existen gran cantidad de servidores especializados en el mercado para los servicios mencionados arriba, con altas prestaciones para multiprocesamiento y redundancia. El precio de estos equipos muchas veces implica grandes inversiones; además, cuando una máquina de este tipo queda obsoleta, las compañías que la tienen se limitan a reemplazar el equipo por uno nuevo. Por ello, en el presente artículo se presentan algunas ideas de cómo construir y configurar un clúster de balanceo de carga para todo tipo de servicio por internet, de manera rápida, íntegra e ininterrumpida, y a su vez, accesible por su licencia y bajo costo. Asimismo, se exponen los componentes de un clúster, su funcionamiento, sus ventajas y desventajas y los resultados obtenidos por su uso.(p. 93)

El comienzo del término y del uso de este tipo de tecnología es desconocido pero se puede considerar que comenzó a finales de los años 50 y principios de los años 60. La historia de los primeros grupos de computadoras es más o menos directamente ligada a la historia de principios de las redes, como una de las principales motivaciones para el desarrollo de una red para enlazar los recursos de computación.

Utilizando el concepto de una red de conmutación de paquetes, el proyecto logró crear en 1969 lo que fue posiblemente la primera red de computadoras básicas basadas en el clúster de computadoras por cuatro similar a un "clúster" pero no un "comodity clúster" como hoy en día lo entendemos. El primer producto comercial de tipo cluster fue Datapoint pero no obtuvo un éxito comercial y los clústeres no consiguieron tener éxito hasta que en 1984 VAXcluster produjeran el sistema operativo VAX/VMS.

Segun Buyya, R. **High Performance Clúster Computing: Programing and applications.** Volumen 2. Prentice-Hall PRT. (1999).

“Un Clúster no es más que un conjunto de computadoras construidas mediante la utilización de componentes de hardware que se comportan como si fuesen una única computadora. La tecnología de clúster ha evolucionado gracias al apoyo de actividades que van desde aplicaciones de súper cómputo, software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos”.

El cómputo con clúster surge como resultado de la convergencia de varias tendencias actuales. Incluye disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, desarrollo de herramientas de software para cómputo distribuido de alto rendimiento y la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Clúster de Alto Rendimiento: Son clúster en los cuales se ejecutan tareas que requieren una gran capacidad computacional, cantidades enormes de memoria o ambas a la vez. Llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos.

**2.2.2.1. Clúster de Alta Disponibilidad**:Son clúster cuyo objetivo es proveer disponibilidad y confiabilidad. Estos clúster tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante un software que detecta fallos y permite recuperarse frente a ellos, mientras que en hardware se evita tener un único punto de fallos.

**2.2.2.2. Clúster de Alta Eficiencia:** Son clúster cuyo objetivo de diseño es ejecutar la mayor cantidad de tareas en el menor tiempo posible; existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

**2.2.2.3. Balanceo de carga:** Clúster que permite que un conjunto de servidores compartan la carga de trabajo y de tráfico a sus clientes. Está compuesto por uno o más ordenadores (llamados nodos) que actúan como front-end del clúster y se ocupa de repartir las peticiones de servicio que reciba el clúster a otros ordenadores que forman su back-end.

**2.2.3 Git**

Es un sistema de control de versiones distribuido, el cual no depende de acceso a la red o de un repositorio central, enfocado a la velocidad, es de uso práctico y manejo de proyectos grandes. Creado por Linus Torvalds, creador del núcleo Linux, para apoyar el desarrollo del kernel de Linux.

Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos. Esto quiere decir que cada vez que se confirma, o se guarda el estado de un proyecto en Git, básicamente toma una fotografía de lo que todos los archivos y almacena una referencia a esa instantánea. Para ser eficaz, si los archivos no tienen cambió, Git no almacena el archivo de nuevo, sólo un enlace al archivo idéntifica al anterior que tiene ya almacenada.

Tambien cuenta con tres estados principales que sus archivos pueden residir en: cometido, modificación y puesta en escena. Medios comprometidos significa que los datos no seguros son almacenados en su base de datos local. Cuando se habla de modificado significa que si se ha cambiado el archivo, pero tienen no comprometidos a la base de datos todavía. Por etapas significa que se ha marcado una modificación del archivo en su versión actual para entrar en la siguiente instantánea.

**2.2.4. Lenguaje de Programación**

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Se define como un idioma artificial diseñado para describir el conjunto de acciones consecutivas que un computador debe ejecutar. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Se definen y construyen sobre los principios de algunas herramientas matemáticas, a fin de permitir dar solución a problemas particulares en el área sobre la cual se fundamenta el lenguaje. Todo lenguaje de programación está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Un lenguaje de programación es muy estricto: A CADA instrucción le corresponde UNA acción de procesador.

**2.2.5. Lenguaje Etiqueta**

Un lenguaje etiqueta ocurre cuando cada paquete de información está delimitado por dos etiquetas como se hace también en el lenguaje HTML, pero XML separa el contenido de la presentación Las etiquetas son piezas de texto encerradas ente los signos menor que (<) y mayor que (>), es una marca con clase que delimita una región en los lenguajes basados en XML. La Web se basa en el HTML, o «lenguaje de marcado de hipertexto», que está basado en el uso de etiquetas. Las etiquetas le dicen al programa visualizador de páginas web (o navegador) en qué juego de caracteres está la página, de qué tipo es cada uno de los fragmentos de texto que contiene , si están alineados a un lado o centrados, en qué tipo de letra está el texto, si hay tablas, de qué anchura son , entre otros. Dicho de otro modo: las etiquetas dan al navegador las instrucciones necesarias para que presente la página en pantalla.

**2.2.6. HTML5**

(HyperText Markup Language, versión 5), es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).1 2 Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

“Entender HTML5 es entender que hoy se conectan desde teléfonos móviles, tabletas, eBooks, netbooks, computadores y otra gama de dispositivos. Es entender que se acabaron los webmasters y hoy hablamos de equipos multidiciplinarios de empresas de tecnología que cuentan con frontends, backends, sysadmins, mobile devs, comunity managers y arquitectos de información en los proyectos que están reiventando mercados y generando tráfico e ingresos”. Vega, Van Der Henst (2011).

**2.2.7. CSS3**

CSS son las siglas de “Hoja de estilo en Cascada”, que sirve para darle estilo a la estructura creada en HTML. CSS3 es la última versión hasta la fecha y presenta como principales características, mayor control sobre el estilo de los elementos de una página web y mayor número de efectos visuales. Una de sus ventajas es que brinda un código más simple para algunas tareas y mayores opciones de gráfica. CSS3 está dividida en varios documentos separados, llamados "módulos". Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

**2.2.8. Less**

LESS (Leaner CSS) es un Framework CSS que proporciona una alternativa para SASS. Se extiende CSS con variables mixins, operaciones y reglas anidadas. Como una extensión al CSS, LESS no es sólo compatible con CSS, pero las características adicionales que añade utilizan la sintaxis CSS. Se puede escribir código LESS igual que lo haría escribir CSS, con la excepción que necesita para compilarlo para CSS. Ahí es donde se puede utilizar LESS Rubí en vez de compilar LESS en código CSS. Teniendo en cuenta que puede utilizar LESS compilador para traducir el archivo de LESS, en un archivo CSS, y usarlo en cualquier tipo de proyecto.

Se analiza de forma automática las aplicaciones de archivos .less a través LESS y envía los archivos CSS. De manera recurrente se ve por LESS (less) los archivos en app / hojas de estilo, guarda los archivos CSS resultantes a hojas de estilo públicos usando la misma estructura de directorios app / hojas de estilo.

**2.2.9. Twitter Bootstrap**

Es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript.

Bootstrap primero se diseñó más como una guía de estilo para documentos no sólo con nuestras características, pero mejora la práctica y la vida. Es de código abierto, soporta diseños sensibles, esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).

**2.2.10. Framework**

Es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar.

**2.2.11. Cherrypy**

CherryPy es un framework de desarrollo web orientado a objetos en Python, provee las bases sobre las cuales aplicaciones web complejas pueden ser escritas, con poco conocimiento. Permite a los desarrolladores construir aplicaciones web en una forma muy similar a la que construirían cualquier programa Python, esto generalmente resulta en menos tiempo de desarrollo. Las aplicaciones CherryPy son usualmente muy simples, el comportamiento por defecto es lo suficientemente sensible para permitir un uso sin configuraciones o personalizaciones extensas. El servidor web embebido, permite a uno desplegar aplicaciones web en donde Python esté instalado.

CherryPy ha empezado a utilizar Los puntos fuertes de Python como un lenguaje dinámico para modelar y unen el protocolo HTTP en una aplicación que sigue expresiones Python, viene con su propio servidor web (HTTP). El objetivo de esta decisión fue la de hacer CherryPy autónomo y permitir a los usuarios ejecutar una aplicación CherryPy minutos después de llegar a la biblioteca. Viene con su propio sistema de configuración que le permite parametrizar el servidor HTTP, así como el comportamiento del motor de CherryPy al procesar una Request-URI y un conjunto de módulos que cubren las tareas comunes en la construcción de una red aplicación, como la gestión de sesiones, el servicio de recursos estáticos, manejo de codificación, o caché de base.

**2.2.12. NoSQL**

Sistemas de gestión de bases de datos relacionales (RDMBS o Relational Database Management System), hoy en día son la tecnología predominante para almacenar datos estructurados en aplicaciones web y empresariales. NoSQL ó como suele llamarse “No solo SQL”, es una filosofía de sistemas de gestión de bases de datos que modifican por completo el modelo clásico de bases de datos relacionales (RDBMS).

Esta nueva forma de trabajar responde a otra forma de organización de los datos que permiten una menor rigidez de los datos y formas novedosas de trabajo, entre sus características comunes de implementaciones de bases de datos distribuidas no relacionales o NoSQL, proporcionan un flujo de datos mucho mayor que los tradicionales RDBMS, ofrece escalabilidad horizontal y ejecución en hardware, ya que a diferencia de los sistemas de gestión de bases de datos relacionales, las bases de datos NoSQL están diseñados para escalar bien en la dirección horizontal y no se basan en hardware de alta disponibilidad.

Las máquinas se pueden agregar y quitar sin causar los mismos esfuerzos operativos para realizar un fragmento de base de datos en RDBMS, algunos almacenes de datos NoSQL incluso proporcionar fragmentos de base de datos automático. Disminuye la tolerancia a fallos (debido a la estructura ACID), redundancia y sin cuellos de botella, cuenta con un escalamiento sencillo, donde el costo de implementación es menor comparado con las base de datos relacionales.

También cuenta con una serie de desventajas relevantes, como lo son el evitar la implementación de bases de datos relacionales, ya que no utiliza SQL como el principal lenguaje para realizar consultas, resultando una complejidad innecesaria proporcionar una variedad de características y datos estrictos de consistencia, ocupando un gran tiempo de procesamiento de consultas. Así como no garantizar completamente ACID (atomicidad, coherencia, aislamiento y durabilidad), ni garantizar operaciones JOIN, que no es más que permitir combinar registros de dos o más tablas en una base de datos relacional. Existen diferentes tipos de bases de datos NoSQL, siendo los principales:

**2.2.12.1. Base de Datos Documentales (basadas en documentos):** Estas base de datos está constituida por un conjunto de programas que almacenan, recuperan y gestionan datos de documentos o datos de algún modo estructurados; constituyen una de las principales subcategorías dentro de las denominadas bases de datos no SQL. A diferencia de las bases de datos relacionales, estas bases de datos están diseñadas alrededor de una noción abstracta de "Documento". Cada implementación de base de datos orientada a documentos difiere en los detalles, en general todas ellas comparten el principio de que los documentos encapsulan y codifican datos o información siguiendo algún formato estándar. Algunos ejemplos de implementación de esta base de datos son: CouchDb, MongoDB y IBM Lotus,RavenDB, SimpleDB.

**2.2.12.2. Bases de Datos en Grafos:** Estas base de datosrepresentan la información como nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos ya que esta puede describir atributos de los nodos (entidades) y las aristas (relaciones). Una base de datos orientada a grafos debe estar absolutamente normalizada, esto quiere decir que cada tabla tendría una sola columna y cada relación tan solo dos, con esto se consigue que cualquier cambio en la estructura de la información tenga un efecto tan solo local, siguiendo un camino de grafo para conseguir la información. Algunos ejemplos son: Neo4j y AllegroGrap, InfiniteGrap, InfoGrid.

**2.2.12.3. Bases de Datos Clave-Valor:** Es la forma más típica, donde cada elemento está identificado por una llave única, lo que permite la recuperación de la información de manera muy rápida. Normalmente el valor se almacenar como un objeto BLOB. De esta forma el tipo de contenido no es importante para la base de datos, solo la clave y el valor que tiene asociado. Son muy eficientes para lecturas y escrituras, además de que pueden escalar fácilmente particionando los valores de acuerdo a su clave. Algunos ejemplos son: Cassandra, BigTable, Dynamo, Redis.

**2.2.12.4. Base de Datos Tabulares:** Este tipo de base de datosguardan los valores en columnas en lugar de filas. Con este cambio ganamos mucha velocidad en lecturas, ya que si se requiere consultar un número reducido de columnas, es muy rápido hacerlo pero no es eficiente para realizar escrituras. Por ello este tipo de soluciones es usado en aplicaciones con un índice bajo de escrituras pero muchas lecturas. Algunos ejemplos: Hbase, BigTable, LevelDB,HyperTable.

**2.2.13. MongoDB**

Es una base de datos NoSQL orientado a documentos, no una relación, desarrollado bajo el concepto de código abierto. La razón principal para alejarse del modelo relacional es que hacer escala fuera más fácil, pero hay algunas otras ventajas también. La idea básica consiste en sustituir el concepto de una "fila", con un modelo más flexible, el "Documento", el documento orientado enfoque hace posible representar relaciones jerárquicas complejas. Esto encaja muy naturalmente en la forma, en que los desarrolladores de lenguajes orientados a objetos modernos piensan en sus datos, cuenta con drivers para los lenguajes de programación bajo la licencia Apache.

MongoDB fue diseñado desde el principio para escalar. Sus datos orientados a documentos modelo permite dividir automáticamente los datos entre varios servidores. Se puede equilibrar datos y la carga en un clúster, redistribuyendo documentos automáticamente. Esto permite los desarrolladores centrarse en la programación de la aplicación, y no su reducción. Usa tipos de datos de JSON, almacenando los datos en formato BSON, soporta índices secundarios genéricos, lo que permite una variedad de consultas rápidas, las cuales son soportadas por Ad Hoc (por campos, rangos y expresiones regulares) y proporcionando de manera única y compuesta, en lugar de los procedimientos almacenados, los desarrolladores pueden almacenar y utilizar las funciones de JavaScript como lenguaje para sus líneas de comando y almacenarlas en documentos, utilizando la tecnología GridFS, ejecutando los valores en el lado del servidor.

Algunas de las características comunes a bases de datos relacionales no están presentes en MongoDB. Estas son decisiones arquitectónicas para permitir la escalabilidad, porque tanto de esas características son difíciles de proporcionar de manera eficiente en un sistema distribuido

El desarrollo de MongoDB empezó con la empresa de software 10gen en el 2007, cuando estaban desarrollando una plataforma como servicio (PAAS) similar al conocido Google App Engine. En el 2009 MongoDB fue lanzado como un producto independiente y publicado bajo la licencia de código abierto AGPL. En marzo de 2011, se lanzó la versión 1.4 y se consideró ya como una base de datos lista para su uso en producción.

**2.3 Definición de Términos Básicos**

**Ambiente de Programación Paralela:** Permiten implementar algoritmos que hagan uso de recursos compartidos: CPU (central processing unit ó unidad central de proceso), memoria, datos y servicios.

**Apache:** Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, entre otros), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

**BSD (**Berkeley Software Distribution, Distribución de Software Berkeley)**:** Es un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

**Distribuciones:** También conocido como software distro, es una recopilación de software específico o una colección de múltiple software, incluso un sistema operativo, ya compilado y configurado preparado para su instalación.

**DNS (**Domain Name System, Sistema de Nombres de Dominio)**:** Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada, que asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

**FTP** (File Transfer Protocol, Protocolo de Transferencia de Archivos)**:** Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basada en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

**HTTP** (Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto)**:** Es el protocolo usado en cada transacción de la World Wide Web. No guarda ninguna información sobre conexiones anteriores, para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

**Hardware:** Se refiere a todas las partes tangibles de un sistema informático.

**Linux:** Es un núcleo libre de sistema operativo basado en Unix, es uno de los principales ejemplos de software libre. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo.

**Middleware:** Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, entre otros.

**Nodo:** Es cada uno de los computadores individuales que forman un sistema multicomputador.

**Nodo Activo/Activo:** Es cuando todos los servidores del clúster pueden ejecutar los mismos recursos simultáneamente. Es decir, los servidores poseen los mismos recursos y pueden acceder a estos independientemente de los otros servidores del clúster. Si un nodo del sistema falla y deja de estar disponible, sus recursos siguen estando accesibles a través de los otros servidores del clúster.

**Nodo Activo/Pasivo:** Consiste en un servidor que posee los recursos del clúster y otros servidores que son capaces de acceder a esos recursos, pero no los activan hasta que el propietario de los recursos ya no esté disponible.

**OpenSSH** (Open Secure Shell)**:** Es un conjunto de aplicaciones que permiten realizar comunicaciones cifradas a través de una red, usando el protocolo SSH.

**Objeto BLOB (**Binary Large Objects, objetos binarios grandes): Son elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica. No todos los Sistemas Gestores de Bases de Datos son compatibles con los BLOB. Generalmente, estos datos son imágenes, archivos de sonido y otros objetos multimedia.

**Protocolos de comunicación y servicios:** Definen las normas que permiten que se establezca una comunicación entre varios equipos o dispositivos, ya que estos equipos pueden ser diferentes entre sí y manejar distintos lenguajes.

**Panel de Control:** Es una herramienta que permite modificar diversos parámetros en la configuración de un software o de un hardware.

**Sistema Operativo:** Es un conjunto de programas que controla la ejecución de programas de aplicación y actúa como una interfaz entre el usuario y el hardware de una computadora, es decir un Sistema Operativo explota y administra los recursos de hardware de la computadora con el objeto de proporcionar un conjunto de servicios a los usuarios del sistema. Existen sistemas operativos basados en diferentes plataformas, entre ellos Windows, Linux, Unix, Mac.

**Software:** Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación.

**SSH (**Secure SHell, intérprete de órdenes segura)**:** Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

**SCP** (Secure Copy)**:** Es un medio de transferencia segura de archivos informáticos entre un host local y otro remoto o entre dos hosts remotos, usando el protocolo Secure Shell (SSH).

**CAPÍTULO III**

**MARCO METODOLÓGICO**

**3.1 Tipo de Investigación**

El presente proyecto se ubica en la modalidad de tipo documental y factible ya que es una propuesta viable destinada a satisfacer necesidades específicas a partir de un diagnóstico. Apoyada en una investigación de campo, en el manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales de la Universidad Pedagógica Experimental Libertador (UPEL, 2006) que se define como:

El proyecto factible consiste en la investigación, elaboración y desarrollo de una propuesta, de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos; puede referirse a la formulación de políticas, programas, tecnologías, métodos, o procesos. (p. 13).

**3.2 Diseño de la Investigación**

Como se menciona anteriormente, este proyecto se apoya en una investigación de tipo descriptivo, documental y de campo, ya que según Zorrilla (1993), expresan lo siguiente:

La investigación documental es aquella que se realiza a través de la consulta de documentos (libros, revistas, periódicos, memorias, anuarios, registros, códices, constituciones, etc.). La de campo o investigación directa es la que se efectúa en el lugar y tiempo en que ocurren los fenómenos objeto de estudio. La investigación mixta es aquella que participa de la naturaleza de la investigación documental y de la investigación de campo. (p. 43)

Dicho esto, se considera que este Proyecto Factible apoya su investigación en un diseño de campo y documental. Ya que los datos para su realización fueron obtenidos directamente del lugar donde se basó la investigación y de otros muchos de material teórico ya existente. El presente trabajo se enmarca en la modalidad de proyecto factible, pues se plantea la solución de un problema real, que presenta mejoras a proyectos ya existentes y componentes nuevos para dar solución a dicho problema.

**3.3 Nivel de la Investigación**

La presente investigación es de tipo descriptiva, ya que se miden y evalúan diversos aspectos, dimensiones y componentes de un sistema. La investigación descriptiva según la UPEL (2006)

"Es el tipo de investigación concluyente que tiene como objetivo principal la descripción de algo, generalmente las características o funciones del problema en cuestión" (p. 9)

**3.4 Fases Metodológicas**

En esta sección se exponen los pasos que se emplearon para dar repuestas a los objetivos planteados. Se especifican las herramientas usadas en cada fase de la investigación, y se presentan los pasos que amerita el desarrollo con la metodología de Desarrollo Guiado a Pruebas (TTD), llamada también Test-Driven Development, derivando de allí sus siglas en inglés, siendo una técnica iterativa de diseño e implementación de software, donde cada nueva línea de código se escribe en respuesta a una prueba fallida que el propio programador ha desarrollado. Es parte medular del desarrollo ágil de código derivado de la metodología XP, donde resulta en una práctica crítica, y de los principios del Manifiesto Ágil. Su principal objetivo es obtener código limpio que funcione.

La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas sean exitosas se garantizará que el software cumple con los requisitos que se han establecido. TDD puede ser aplicado a dos niveles, el Nivel de micro-iteraciones, el cual es guiado por pruebas unitarias escritas por el propio programador de la aplicación, se realiza mediante tres fases: Escribir y ejecutar pruebas, escritura de código, y la refactorización (Refactoring). El siguiente es el Nivel iteración o funcional, el cual es guiado por pruebas de aceptación. Estas fases se realizan de manera recursiva hasta lograr el objetivo deseado.

**Fase I: Escritura y ejecución de pruebas**

En esta fase, se describen los distintos requerimientos y pruebas unitarias a implementar para lograr un buen funcionamiento. Esas pruebas se describieron mediante la codificación de los módulos base como lo son, el módulo de control de usuarios del sistema, control de servicios, de manejo del shell, control de los programas/paquetes, control de los archivos de configuración, entre otros; siendo base primordial para el funcionamiento del sistema. Esta codificación consta de las funciones básicas de cada módulo, pero no las necesarias para su funcionamiento, tomando en cuenta así, que a la hora de hacer la verificación la prueba falle y se pueda pasar a la siguiente fase.

**Fase II: Escritura de código**

En esta fase, luego de haber descrito y verificado la falla de las pruebas, se procede a codificar todo lo necesario para que las pruebas unitarias funcionen. Esta codificación fue un poco menos rigurosa y extensa que en la anterior fase, ya que en esta solo se escribe lo necesario para que los módulos pudieran funcionar y pasar la prueba. A medida que ocurren las iteraciones el código se va perfeccionando.

**Fase III: Refactorización (Refactoring) o Reestructuración**

En esta etapa, se rastrea el código incluyendo el aplicado en la fase I, al buscar líneas duplicadas y eliminarla reestructurando el código, alterando su estructura interna pero sin cambiar su comportamiento externo, así como también se revisa que el código cumpla con ciertos principios de diseño.

**Fase IV:** **Especificación de pruebas de Aceptación**

En esta etapa, se especifican las pruebas antes que las funcionalidades sean implementadas y una vez que el código se escribe la prueba sirve como un criterio de aceptación. Esta metodología permite enumerar las dos primeras fases del ciclo coloreando en rojo aquellas especificaciones que no se cumplen y en verde las especificaciones cumplidas. Cuando se han dado los tres pasos de la especificación que ocupa, se toma la siguiente y vuelve a repetirse.

**CAPÍTULO IV**

**RECURSOS**

Existen varios factores externos como un conjunto de personas, bienes materiales, financieros y técnicos, con que cuenta que condicionan los recursos necesarios para el desarrollo de la investigación, sin embargo esta no presenta una limitante para lograr con éxito la culminación de la misma.

**4.1. Humanos**

Se contará con un conjunto de personas que conforman un valioso recurso disponible para efectuar el estudio, personal que está conformado por el profesor Ing. Luis Llave, el cual se vincula con este proyecto de investigación al proveer su asesoría y conocimientos para la realización del mismo. Así como la profesora metodóloga Ing. Alicia de Pizzella, la cual provee las normas necesarias para su realización y contando con la participación de Irisel González para su construcción. El personal que conforma el recurso humano está disponible para participar en la investigación y con posibilidad de emplear su tiempo para el desarrollo de la misma.

**4.2. Institucionales**

Se recurrió a la Universidad José Antonio Páez en busca de información para realizar del proyecto de investigación.

**4.3. Materiales**

Para la realización de este proyecto se utilizaron los siguientes recursos:

**Software:**

* Python
* Cherrypy
* MongoDB
* Red Hat Enterprise Linux

**Hardware:**

* Computadora
* Impresora
* Así como la utilización de lápiz, papel y fuentes bibliográficas.

**4.4 CRONOGRAMA DE ACTIVIDADES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Actividades** | **Febrero 2013** | **Marzo**  **2013** | **Abril**  **2013** | **Mayo**  **2013** | **Total en semanas** |
| **Escritura y ejecución de pruebas** | **X** |  |  |  | **2** |
| **Escritura de código** |  | **X** |  |  | **2** |
| **Refactorización o reestructuración** |  | **X** |  |  | **3** |
| **Especificación de pruebas de aceptación** |  | **X** | **X** |  | **2** |
| **Generación de documentación necesaria** |  |  | **X** | **X** | **3** |
| **Total** |  |  |  |  | **12** |

**REFERENCIAS.**

**Bibliograficas.**

Buyya, Robert. (1999) .High Performance Clúster Computing: Programing and applications. Volumen 2. Prentice-Hall PRT. Canada.

Chodorow, Kristina, Dirolf, Michael.(2008) MongoDB. The Definitive Guide. 1ra Edición, O’Reilly Media, Inc. United States of America.

Cochran, David. (2012).Twitter Bootstrap Web Development How-To. 1ra Edición. Packt Publishing.

Gift, Noah y Jones Jeremy (2009) Python para administración de sistemas Unix y

Linux, Primera Edición, España, Anaya Multimedia.

Hellegouarch, Sylvain (2007). CherryPy Esencial. Rapid Python Desarrollo de Aplicaciones Web. 1ra Edición. Packt Publishing Ltd. 32. Reino Unido.

Iturriaga, Santiago, Maya, Paulo, Pintos, Damián. (2008). Proyecto Fenton –Clúster de Computadores de Alto Desempeño con Acceso Remoto (CCADAR).Trabajo de ascenso para optar al título de Ingeniero en Computación. Universidad de la Republica. Uruguay.

Loeliger, Jon. (2009). Version Control whit Git. 1ra Edición, O’Reilly Media, Inc. United States of America.

Meza, Antonio (2012). Análisis de Tráfico de Datos en Sistemas Distribuidos. Trabajo de ascenso para Optar al el grado de Maestro en Sistemas Computacionales. Instituto Tecnológico de la Paz. La Paz, Baja California. México.

Montesino, Pedro, Morales, José (2010). Desarrollo de una aplicación web para la administración de Servidores Proxy en Sistemas Operativos Gnu/Linux. Caso: Squid Proxy. Trabajo de ascenso para optar al título de Ingeniero de Computación. Universidad José Antonio Páez. Valencia. Venezuela.

Murillo, Feliz (2000). Introducción a Linux. 1ra Edición. Lima: Centro de Edición del INEI.

Negus, Christopher (2003). Red Hat Linux Biblie:Fedora and Enterprise . 1ra Edition . John Wiley & Sons.

Rocío, Laura, Huezo, Martha, Parada, Octavio. (2009). Análisis del Sistema Operativo Gnu/Linux. Trabajo de ascenso para optar por el título de Licenciado en Ciencias de la Computación. Universidad Centroamericana “José Simeón Cañas”. Antiguo Cuscatlán, el Salvador.

Sinisterra, María, Díaz, Tania, Ruiz, Erik. (2008). Clúster de balanceo de carga y alta disponibilidad para servicios web y mail. 2da Edición. Universidad Libre, Seccional Cali.

Universidad José Antonio Páez. San Diego. (2007). Normas para la elaboración y presentación de los anteproyectos, proyectos y trabajos de grado. Mijares, Héctor.

Universidad Pedagógica Experimental Libertador (2006). Manual de Trabajos de Especialización y Maestría y Tesis Doctorales de la universidad Grado de Pedagógica Experimental Libertador.

**Electrónicas**

Álvarez Miguelangel. Manual de Jquery. Consultado el día 14 de abril de 2013 de la World Wide Web: [http://es.scribd.com/doc/43914919/manual-de-jquery-en-pdf-desarrollowebcom](http://es.scribd.com/doc/43914919/Manual-de-Jquery-en-PDF-Desarrollowebcom).

González, Raúl (2008). Python para Todos, Consultado el día 26 de Marzo de 2013 de la World Wide Web: http://mundogeek.net/tutorial-python/

Lenguaje Etiqueta. Consultado el día 14 de abril de 2013. de la World Wide Web: http://lucas.hispalinux.es/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf

Less. Consultado el día 25 de abril de 2013 de la World Wide Web: <http://lesscss.org/>.

Navajas, Antonio. Guia completa de CSS3. Consultado el día 9 de febrero de 2013 de la World Wide Web: <http://www.antonionavajas.com/blog/categoria-cursos/css3-en-espanol-el-manual-gratis-en-pdf/>.

Pérez, Javier. Introducción a JavaScript.(2009) Consultado el día 20 de Febrero de 2013 de la World Wide Web: <http://www.librosweb.es/javascript>.

Strauch Christof. Consultado el día 23 de abril de 2013 de la World Wide Web: http://www.christof-strauch.de/nosqldbs.pdf.

Vega, John, Van der henst, Christian. Guía HTML5 El presente de la web. Consultado el día 9 de Febrero de 2013 de la World Wide Web: <http://mlw.io/guia-html5/>